



YCCE, Nagpur



MGI

Serial Port interface with 8051

Dr. P. T. Karule

Professor

Department of Electronics Engineering

Yeshwantrao Chavan College of Engineering, Nagpur (M. S.)

Email: ptkarule@ycce.edu

Website: www.ycce.edu

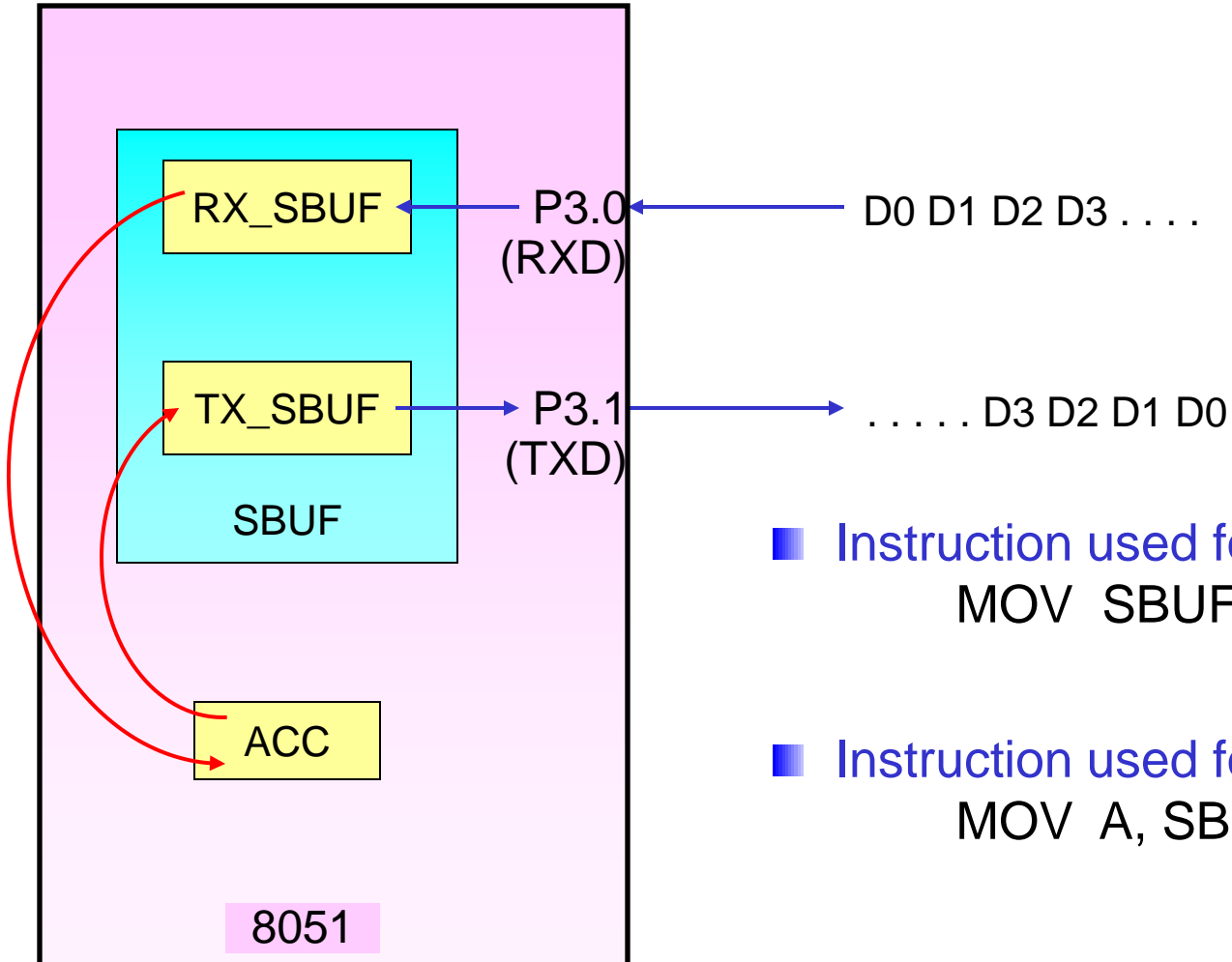
Serial I/O

■ Features

- There are two pins for serial data transfer.
 - ❑ TXD (Transmit data pin)(P3.1)
 - ❑ RXD (Receive data pin)(P3.0)
- There are two SFR's for controlling serial data transfer.
 - ❑ SBUF (Serial Buffer Register) : Used to receive and transmit data, actually there are two serial buffer registers.
 - ❖ TX_SBUF (Transmitter Serial Buffer)
 - ❖ RX_SBUF (Receiver Serial Buffer)
 - ❑ SCON (Serial Control Register): It is used to control the serial data transfer operation.

SFR	Addr	Bit addr
SBUF	99H	--
SCON	98H	98H to 9FH

Serial data transfer



- Instruction used for transmit data
MOV SBUF, A

- Instruction used for receive data
MOV A, SBUF

SCON (Serial Port Control Register)

SCON.7 SCON.6 SCON.5 SCON.4 SCON.3 SCON.2 SCON.1 SCON.0

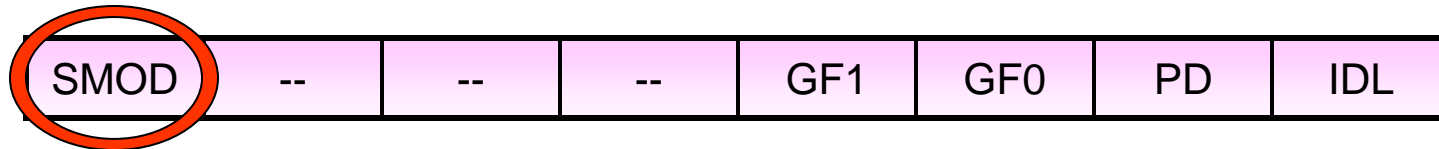
SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM2	If SM2 = 1, enable multiprocessor communication. Normally SM2 = 0
REN	If REN = 1, enable reception. If REN = 0, disable reception.
TB8	The 9 th bit that will be transmitted in mode 2&3 is stored in TB8.
RB8	In mode 2&3 RB8 stores the received 9 th bit. In mode1 it is received stop bit.
TI	Transmit interrupt flag. Set by hardware when complete character is transmitted. i.e. when TX_BUF is empty. Cleared when next byte is transferred to SBUF.
RI	Receive interrupt flag. Set by hardware when complete character is received. i.e. when RX_BUF is full. Cleared when byte is read from SBUF.

SM0	SM1	Mode	Description	Baud Rate
0	0	Mode 0	Shift Register	Fosc/12
0	1	Mode 1	8 bit UART	Variable
1	0	Mode 2	9 bit UART	Fosc/64 or Fosc/32
1	1	Mode 3	9 bit UART	Variable

PCON (Power Control Register)

SFR	Addr	Bit addr
PCON	87H	--



SMOD	Double baud rate bit. If SMOD = 1, the baud rate is doubled when the serial port is used in modes 1,2 or 3.
GF1	General purpose flag bit.
GF0	General purpose flag bit.
PD	Power Down bit. Setting this bit activates power down operation in the 80C51.
IDL	Idle mode bit. Setting this bit activates idle mode operation in the 80C51.

Baud Rates

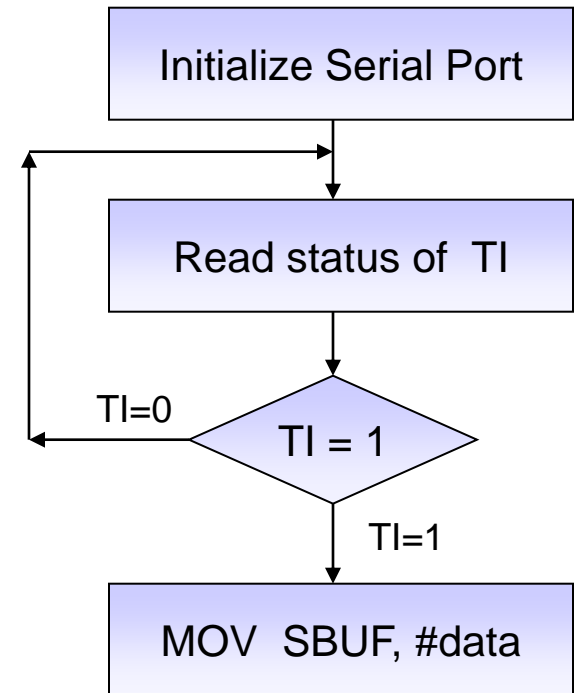
- Following are the commonly used baud rates generated by using **timer 1 in mode2** with count value given in the table.

Baud Rate	Fosc	SMOD	Timer 1		
			C/T	Mode	Reload Value
19.2K	11.059MHz	1	0	2	FDH
9.6K	11.059MHz	0	0	2	FDH
4.8K	11.059MHz	0	0	2	FAH
2.4K	11.059MHz	0	0	2	F4H
1.2K	11.059MHz	0	0	2	E8H
137.5	11.059MHz	0	0	2	1DH

Serial Data Transfer (Tx)

- Define serial data transfer mode & baud rate (Initialization)
 - Initialize SCON for data transfer mode (8 bit data, Variable Baud Rate, Enable Receiver)
 - Initialize TMOD & TH1 for baud rate selection (Timer 1 in Mode2 as Frequency Divider)
 - Start / Run Timer 1 (Set TR1)
- Read & Check Status of **TI = ?**
(check TX_BUF = empty or full)
- **If TI = 1 (TX_BUF is empty) then send data to SBUF.**

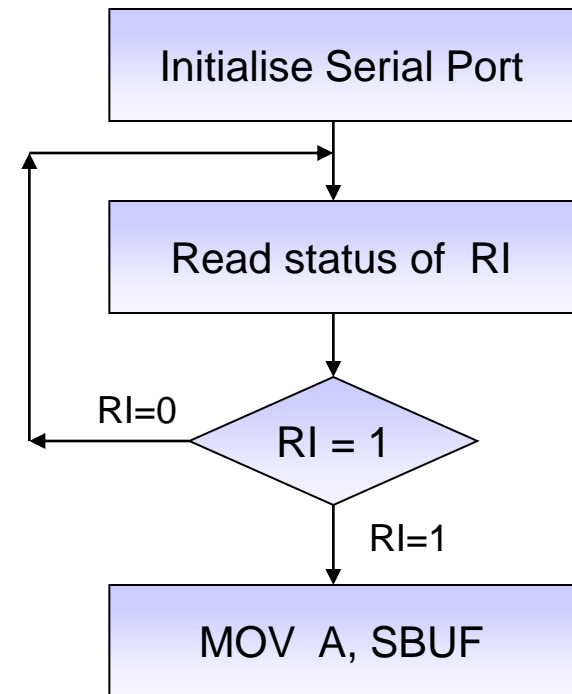
■ Transmit Data



Serial Data Transfer (Rx)

- Define serial data transfer mode & baud rate (Initialization)
 - Initialize SCON for data transfer mode (8 bit data, Variable Baud Rate, Enable Receiver)
 - Initialize TMOD & TH1 for baud rate selection (Timer 1 in Mode2 as Frequency Divider)
 - Start / Run Timer 1 (Set TR1)
- Read & Check Status of **RI = ?**
(check RX_BUF = empty or full)
- **If RI = 1 (RX_BUF is Full) then read data from SBUF.**

■ Receive Data



Program for serial communication

- Write subprogram to initialise serial port (UART) of 8051 with specifications as follows.
 - 8 bit UART
 - baud rate = 9600 (Consider Fosc = 11.059MHz)
- It requires SCON, TMOD, PCON, TH1 to be initialised.

	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
SCON	0	1	0	1	0	0	0	0	= 50H

	gate	c/t	m1	m0					
TMOD	0	0	1	0	0	0	0	0	= 20H

	SMOD	--	--	--	GF1	GF0	PD	IDL	
PCON	0	X	X	X	X	X	X	X	= 00H

Subprograms for serial communication..

.***** UART initialization Routines *****
,

```
SCI_INIT: CLR TR1 ; stop timer1
          MOV SCON, #50H ;8 bit UART,.
          MOV TMOD, #20H ;Timer 1 is set to auto-reload Timer mode.
          MOV TH1, #0FDH ; Baud rate = 9600 for 11.0592 MHz
          SETB TR1 ; Run timer 1
          RET
```

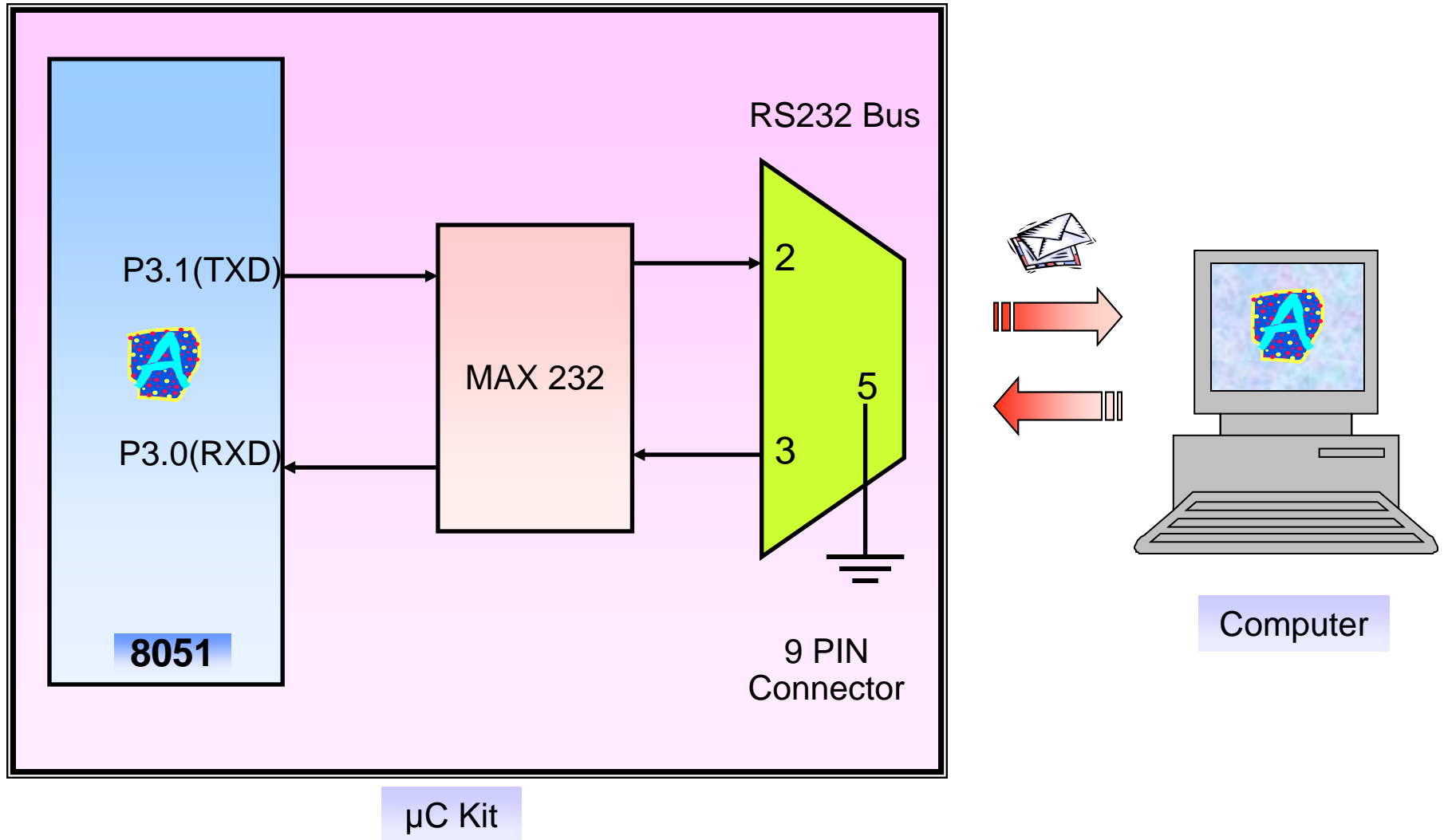
.***** Subroutine to send data *****
,

```
SEND_232: JNB TI, SEND_232 ;Wait until TI =1
          MOV SBUF, A ;Send next byte
          CLR TI ; Clear TI
          RET
```

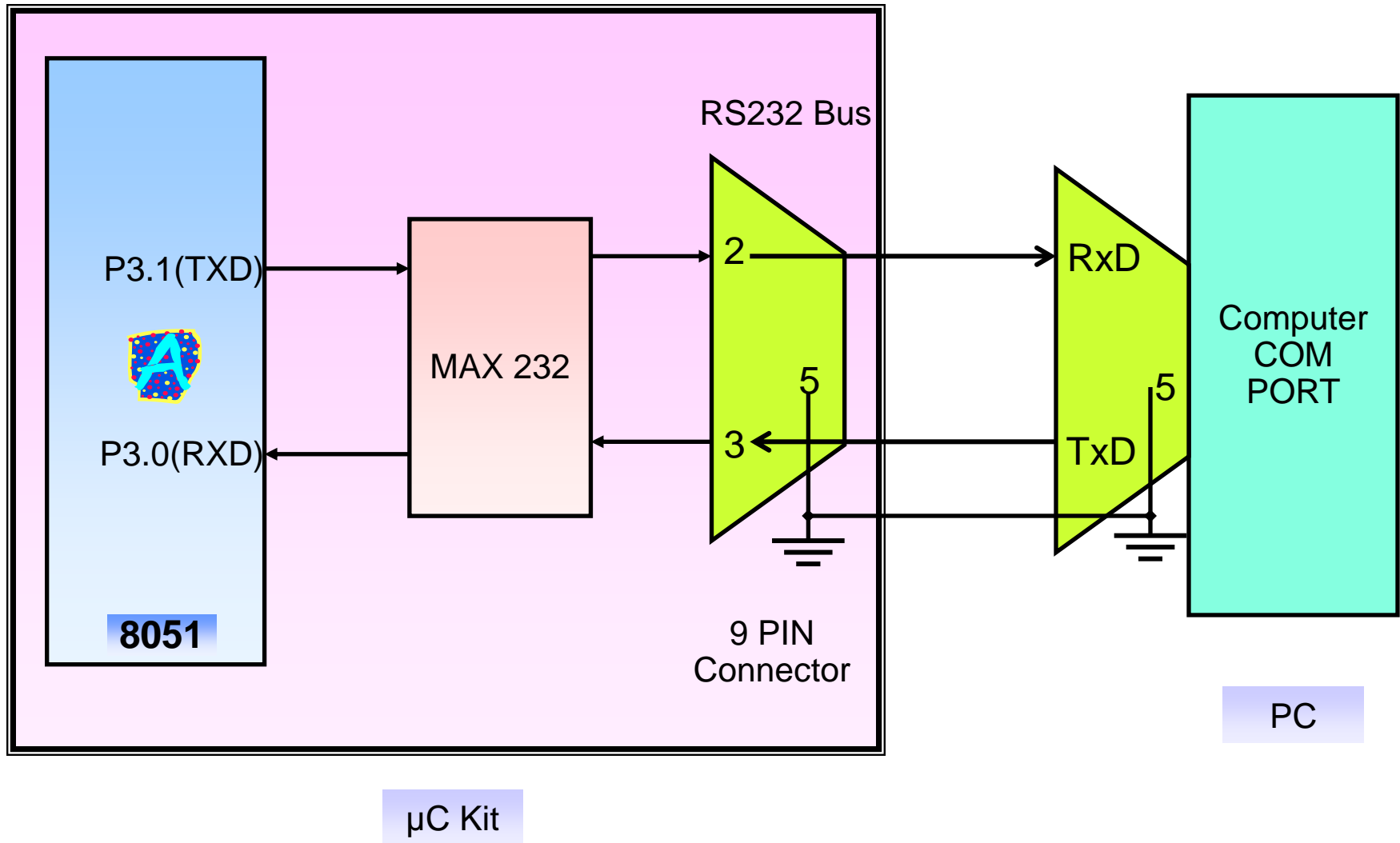
.***** Subroutine to receive data *****
,

```
GET_232: JNB RI, GET_232 ;Wait until character received (RI =1)
          MOV A,SBUF ;Receive data
          CLR RI ;Clear RI
          RET
```

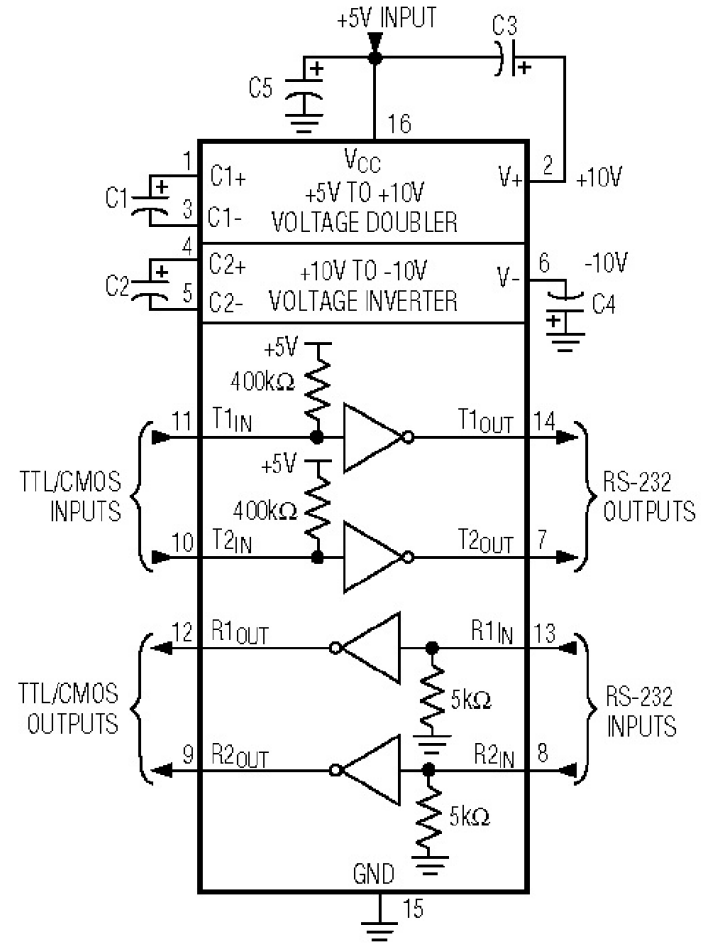
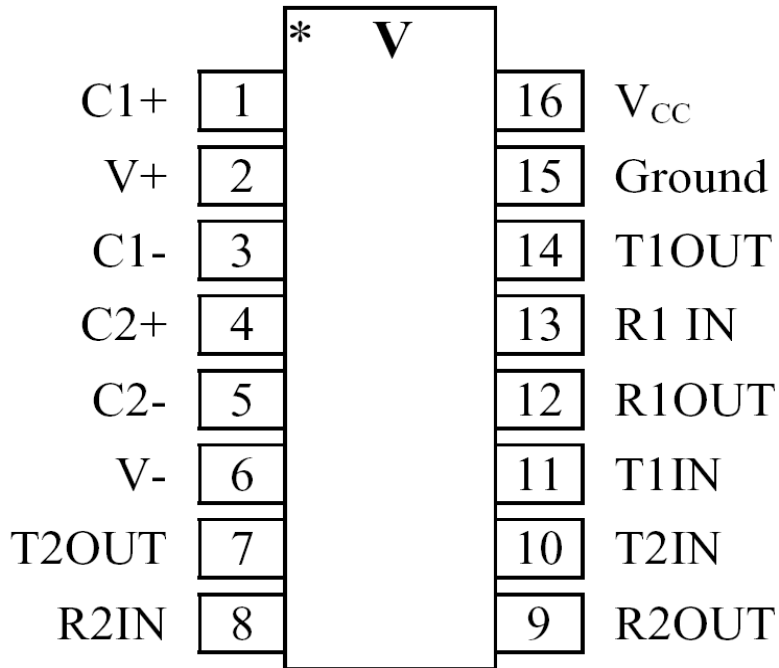
RS232 interface



8051 interface with Computer

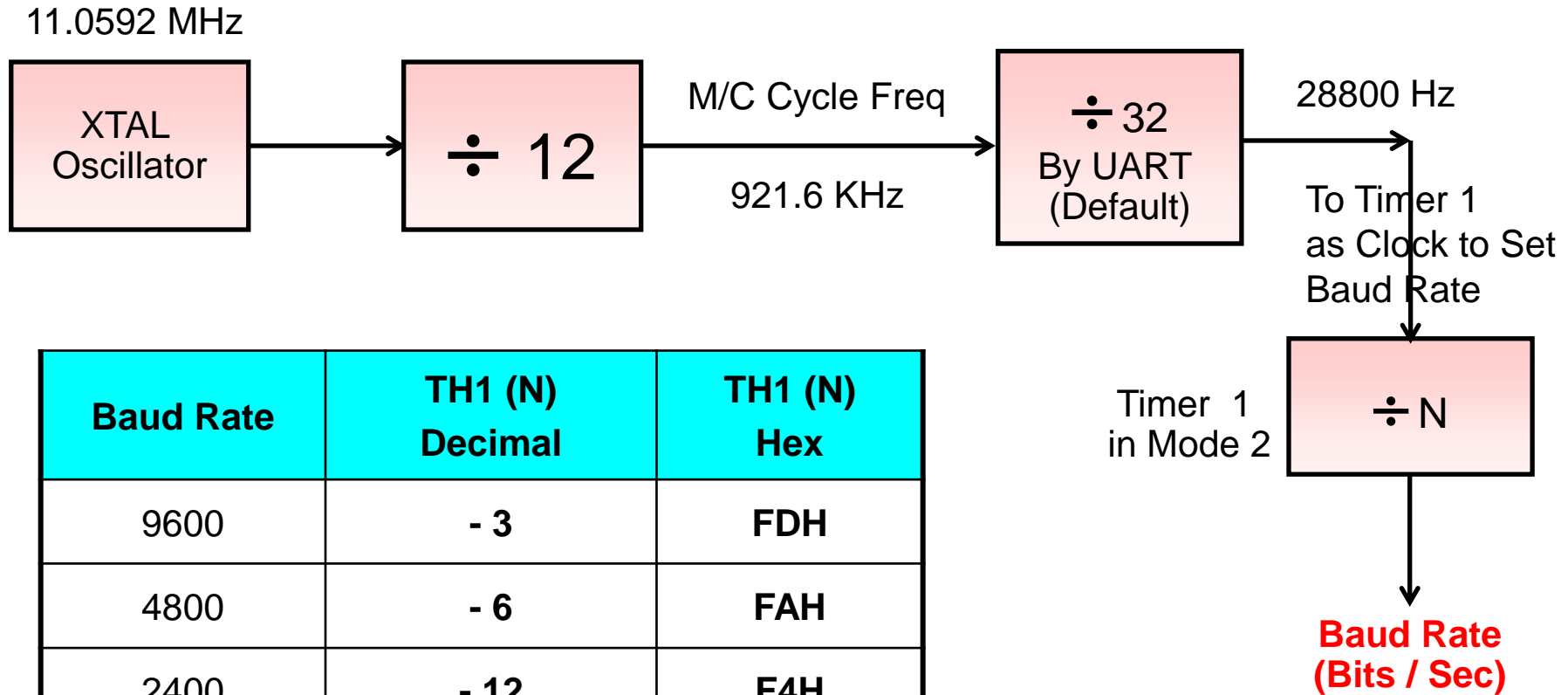


MAX 232



How Baud Rate is Set

- For Baud rate clock is supplied by Timer 1 in Mode 2



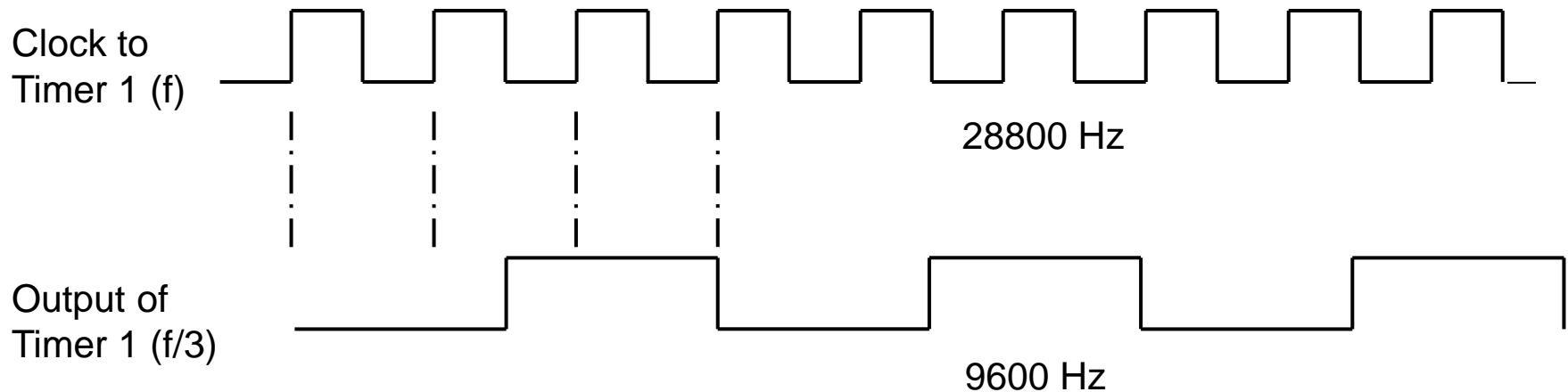
Baud Rate	TH1 (N) Decimal	TH1 (N) Hex
9600	- 3	FDH
4800	- 6	FAH
2400	- 12	F4H
1200	- 24	E8H

How Baud Rate is Set

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

➤ Solution

- The machine cycle frequency of 8051 = $11.0592 / 12 = 921.6$ KHz
- $921.6 \text{ KHz} / 32 = 28,800 \text{ Hz}$ is frequency by UART to timer 1 to set baud rate (by Default UART divide it by 32).
- (a) $28,800 / 3 = 9600$ where $-3 = \text{FD}$ (hex) is loaded into TH1



Example #1

- Write a program for 8051 to transfer letter 'Y' serially with a baud rate of 2400 continuously.

Solution:

- It requires SCON, TMOD & TH1 to be initialized.

SCON

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

= 50H

SM0	SM1	Mode	Description	Baud Rate
0	0	Mode 0	Shift Register	Fosc/12
0	1	Mode 1	8 bit UART	Variable
1	0	Mode 2	9 bit UART	Fosc/64 or Fosc/32
1	1	Mode 3	9 bit UART	Variable

TMOD

gate	c/t	m1	m0
0	0	1	0
0	0	0	0
0	0	0	0

= 20H

TH1 = -12 for 2400 Baud Rate

Example #1...

➤ Program.

Solution:

```
MOV SCON, #50H           ;8-bit, 1 stop, REN enabled
MOV TMOD, #20H           ;timer 1, mode 2(auto reload)
MOV TH1, #-12            ;2400 baud rate
SETB TR1                 ;start timer 1
AGAIN: MOV SBUF, #'Y'     ;letter "Y" to transfer
HERE:  JNB TI,  HERE      ;wait for the last bit
      CLR TI              ;clear TI for next char
      SJMP AGAIN          ;keep sending "Y"
```

Example #2

- Write a program for 8051 to transfer letter 'W' serially with a baud rate of 19200 continuously.

Solution:

- It requires SCON, TMOD, PCON & TH1 to be initialized.

	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	
SCON	0	1	0	1	0	0	0	0	= 50H

	gate	c/t	m1	m0					
TMOD	0	0	1	0	0	0	0	0	= 20H

TH1 = - 3 for 9600 Baud Rate

For 19200 Baud Rate, Use PCON to double the Baud Rate

	SMOD	--	--	--	GF1	GF0	PD	IDL	
PCON	1	X	X	X	X	X	X	X	= 80H

Example #2...

➤ Program.

Solution:

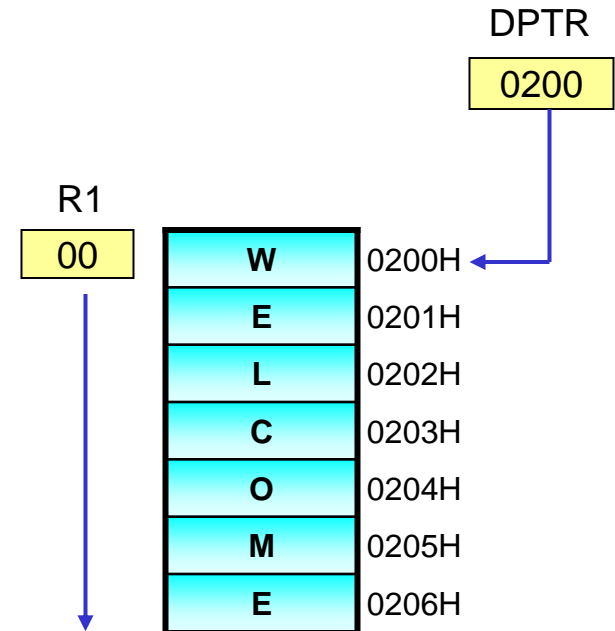
```
MOV SCON, #50H           ;8-bit, 1 stop, REN enabled
MOV TMOD, #20H           ;timer 1, mode 2(auto reload)
MOV TH1, # - 3           ;9600 baud rate
MOV PCON, #80H           ;19200 baud rate
SETB TR1                 ;start timer 1
AGAIN: MOV SBUF, #'W'    ;letter "W" to transfer
HERE:  JNB TI,  HERE     ;wait for the last bit
      CLR TI             ;clear TI for next char
      SJMP AGAIN        ;keep sending "W"
```

Example #3

- Write a program for 8051 to transfer 'WELCOME' serially with a baud rate of 9600, 8 bit UART.

- Program

```
MOV DPTR,#0200H
MOV R1,#00H
MOV SCON, #50H           ;8 bit UART,.
MOV TMOD, #20H          ;Timer 1 in mode2.
MOV TH1,#0FDH           ; Baud rate = 9600
SETB TR1
NEXT: MOV A, R1
      MOVC A,@A+DPTR
L2:   JNB TI, L2          ;Wait until TI =1
      MOV SBUF, A        ;Send next byte
      CLR TI             ; Clear TI
      INC R1
      CJNE R1, #07H, NEXT
L1:   SJMP L1
ORG 0200H
DB 'WELCOME'
END
```

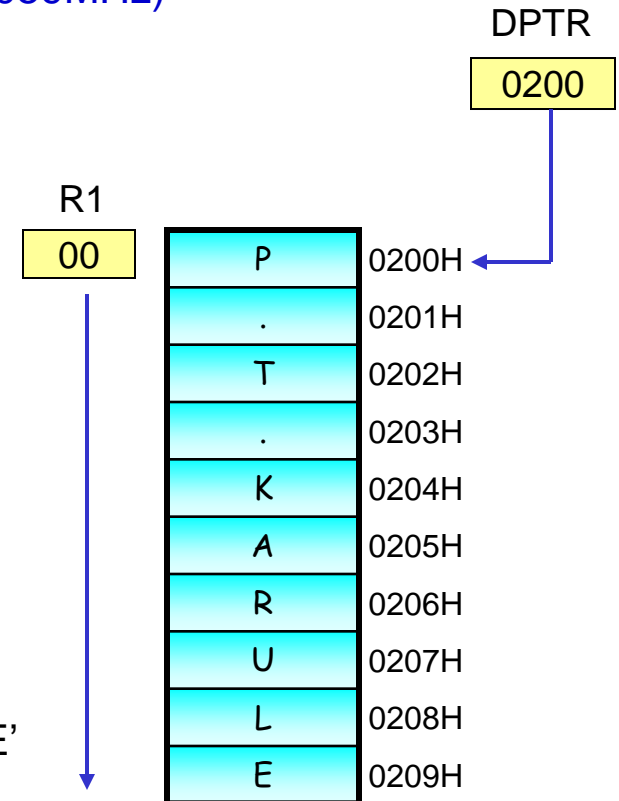


Example #4

- Write program to send message to PC via COM1 port (RS232).
- The message of 10 bytes is present in memory from address 0200H.
- initialize serial port (UART) of 8051 with specifications as follows.
 - 8 bit UART, baud rate = 9600 (Consider Fosc = 11.059MHz)

■ Program

```
$MOD51
ORG 0000H
MOV DPTR, #0200H
MOV R1, #00H
ACALL SCI_INIT
NEXT: MOV A, R1
      MOVC A, @A+DPTR
      ACALL SEND_232
      INC R1
      CJNE R1, #0AH, NEXT
L1:   SJMP L1
ORG 0200H
DB 'P', '.', 'T', '.', 'K', 'A', 'R', 'U', 'L', 'E'
```



Cont...

Example #4..

.***** UART initialization Routines *****
;

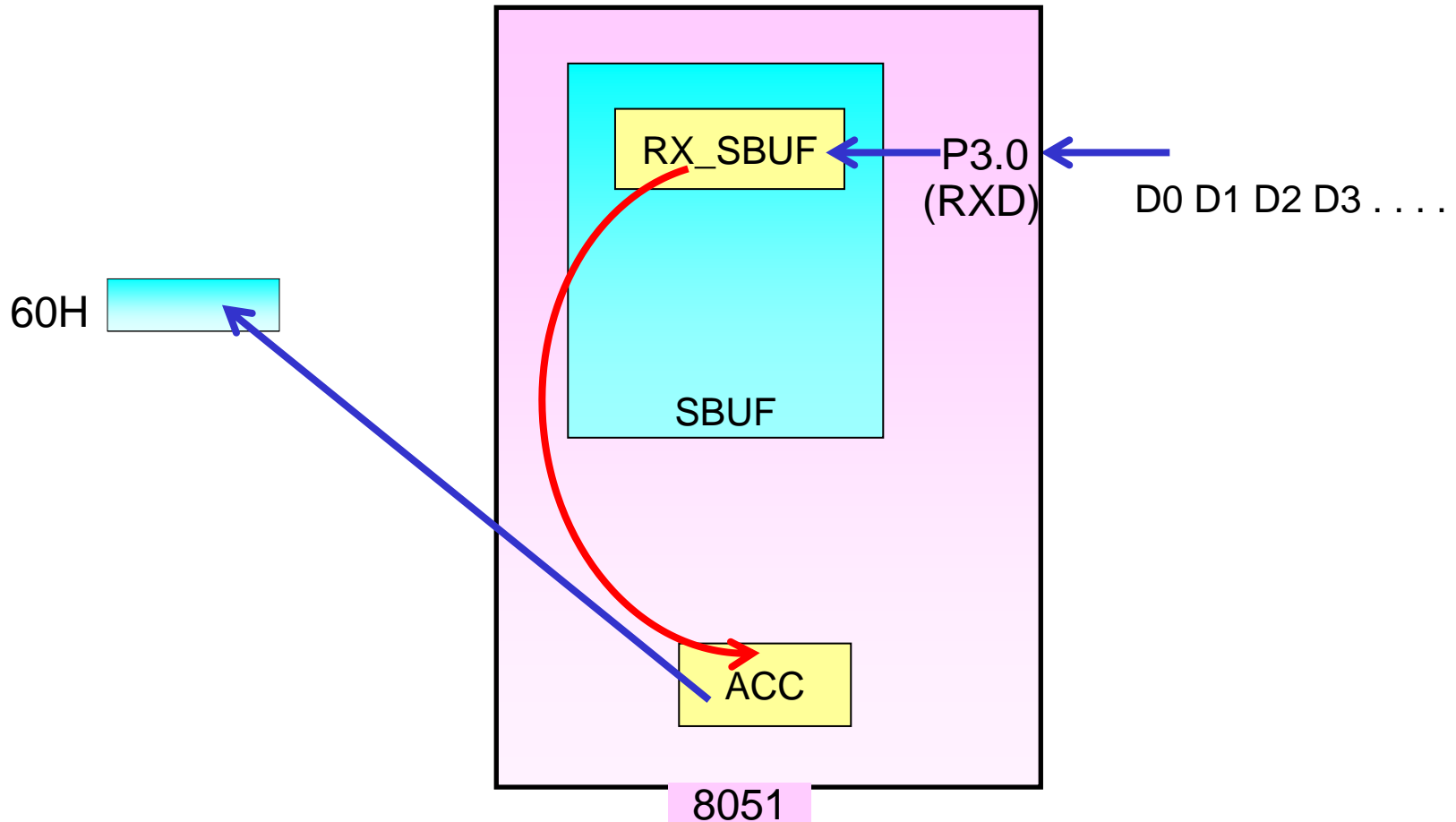
```
SCI_INIT: CLR TR1 ; stop timer1
          MOV SCON, #50H ;8 bit UART, TI set indicates transmitter ready.
          MOV TMOD, #20H ;Timer 1 is set to auto-reload Timer mode.
          MOV TH1, #0FDH ; Baud rate = 9600 for 11.0592 MHz
          SETB TR1 ; Run timer 1
          RET
```

.***** Subroutine to send data *****
;

```
SEND_232: JNB TI,SEND_232 ;Wait until TI =1
          MOV SBUF, A ;Send next byte
          CLR TI ; Clear TI
          RET
          END
```

Example #5

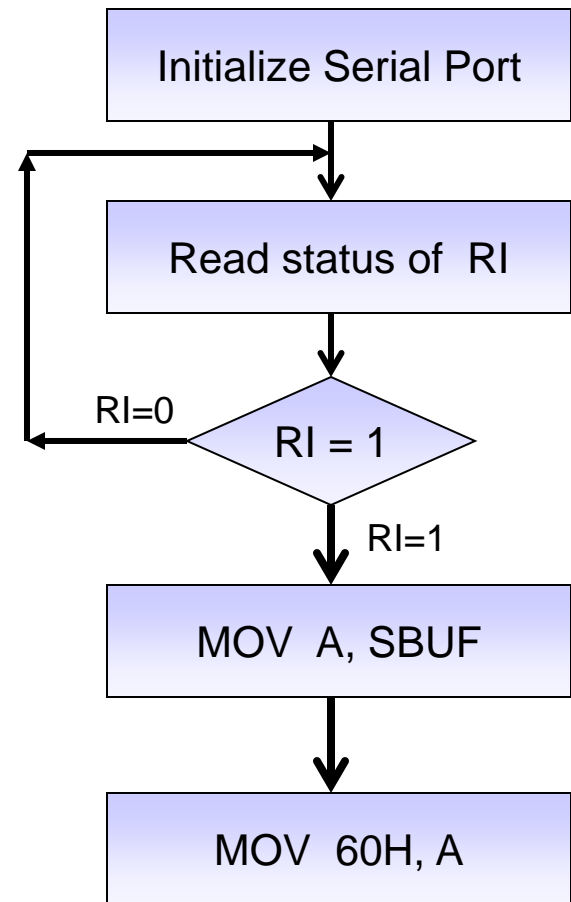
- Write a program to receive data byte serially with 4800 baud rate and save it in memory from address 60 H.



Example #5...

1. Initialize Serial Port
2. Read & Check Status of **RI = ?** (check RX_BUF = empty or full)
3. **If RI = 1 (RX_BUF is Full) then read data from SBUF.**
4. Store data byte in Memory Location 60H
5. Clear RI
6. Stop

■ Flow Chart



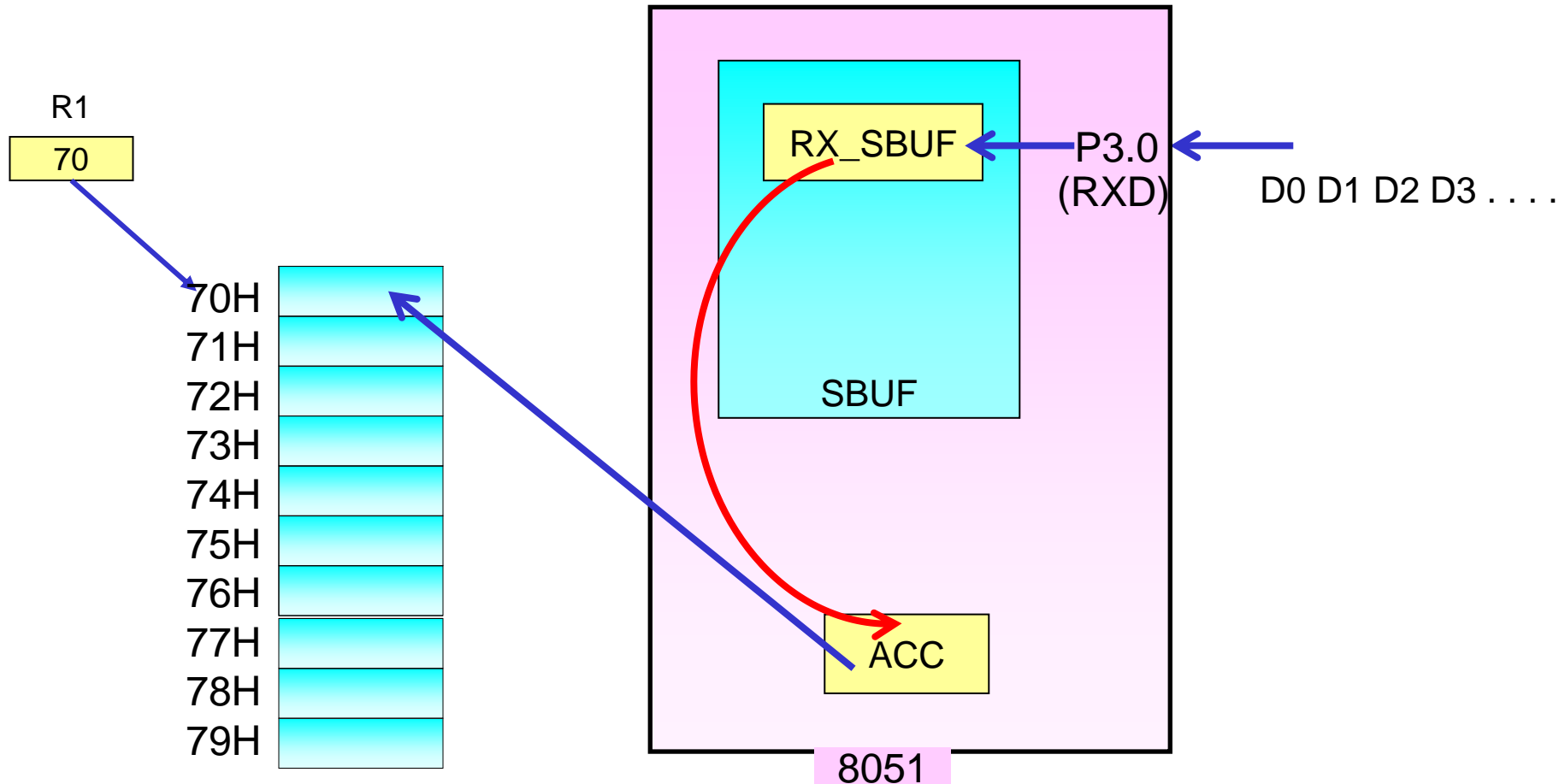
Example #5...

➤ Program

```
MOV  SCON, #50H      ;8 bit UART, Receiver enabled
MOV  TMOD, #20H      ;Timer 1 in mode2.
MOV  TH1,#-6         ; Baud rate = 4800
SETB TR1
WAIT: JNB  RI, WAIT   ;Wait until RI =1
      MOV  A, SBUF     ;Read data byte
      MOV  60H, A      ; Store data byte in M. L. 60H
      CLR  RI         ; Clear RI
L1:   SJMP L1
```

Example #6

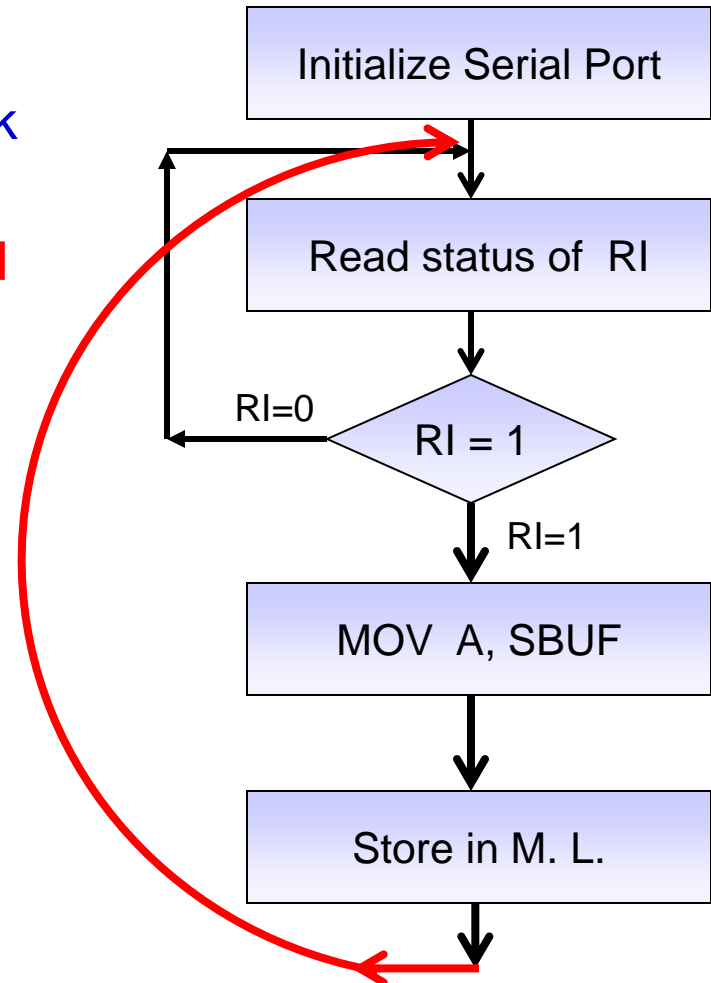
- Write a program to receive 10 data bytes serially with 9600 baud rate. Save the received data bytes in memory from address 70 H. Assume XTAL = 11.0592 MHz.



Example #6...

1. Set Memory Pointer (R1 = 70H)
2. Set Counter (R2 = 0AH)
3. Initialize Serial Port
4. Read & Check Status of **RI = ?** (check RX_BUF = empty or full)
5. **If RI = 1 (RX_BUF is Full) then read data from SBUF.**
6. Store data byte in Memory Location
7. Clear RI
8. Increment Memory Pointer
9. Decrement Counter
10. Repeat step 4 to 9 till counter is 0

■ Flow Chart



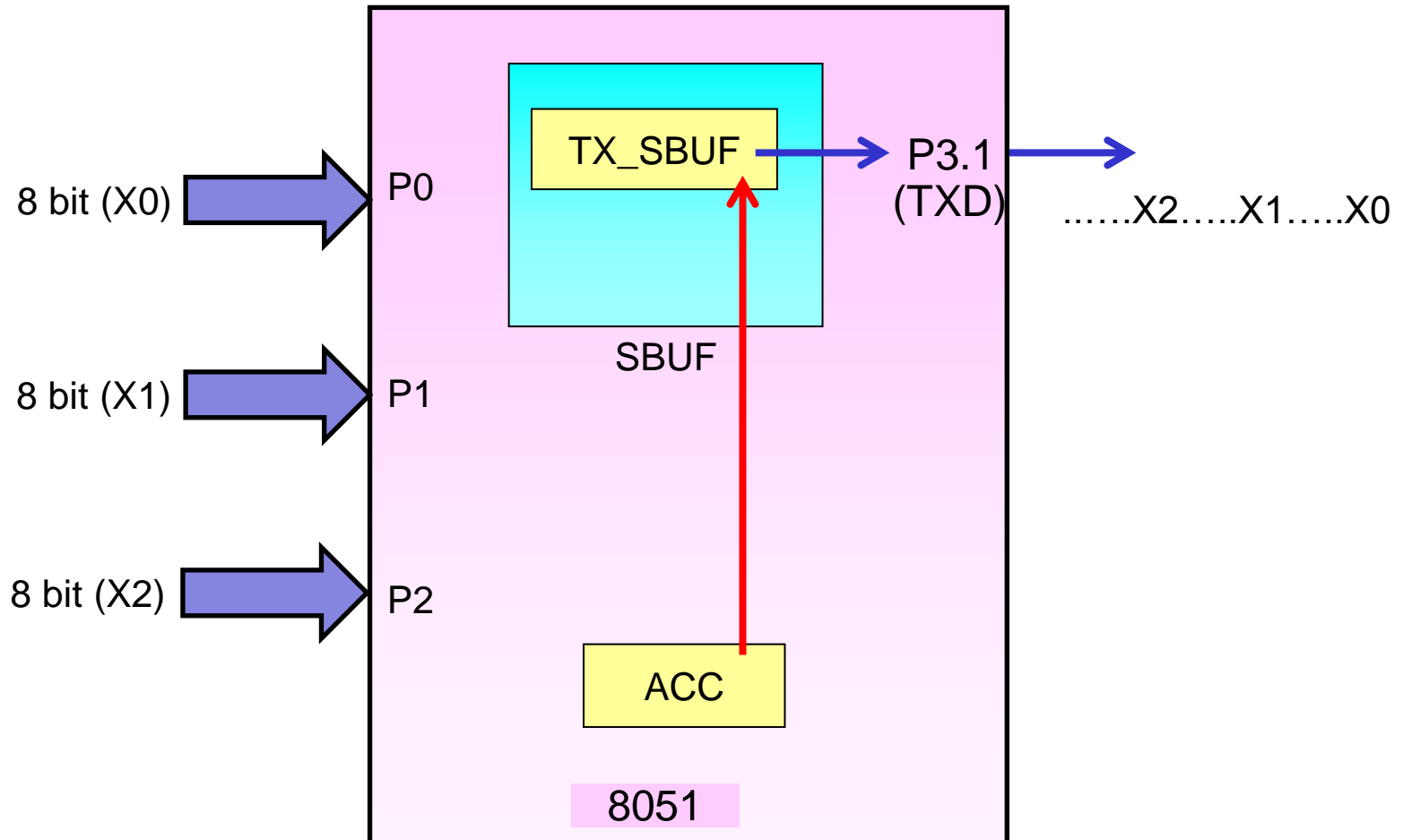
Example #6...

➤ Program

```
MOV R1, #70H           ; Memory Pointer (address 70H)
MOV R2, #0AH           ; Counter for 10 bytes
MOV SCON, #50H         ; 8 bit UART, Receiver enabled
MOV TMOD, #20H         ; Timer 1 in mode2.
MOV TH1, #-3           ; Baud rate = 9600
SETB TR1
WAIT: JNB RI, WAIT      ; Wait until RI =1
MOV A, SBUF            ; Read data byte
MOV @R1, A             ; Store data byte in Memory
CLR RI                 ; Clear RI
INC R1                 ; Increment address
DJNZ R2, WAIT          ; Decrement Counter & Repeat till 0
L1: SJMP L1
```

Example #7

- Write a program to take data in through ports 0, 1 and 2, one after the other and transfer this data serially, with a baud rate of 9600 continuously.



Example #7...

1. Initialize Serial Port
2. Read P0 data
3. Read & Check Status of **TI =?** (check TX_BUF = empty or full)
- 4. If TI = 1 (TX_BUF is Empty) then Send data to SBUF.**
5. Clear TI
6. Read P1 data
7. Read & Check Status of **TI =?** (check TX_BUF = empty or full)
- 8. If TI = 1 (TX_BUF is Empty) then Send data to SBUF.**
9. Clear TI
10. Read P2 data
11. Read & Check Status of **TI =?** (check TX_BUF = empty or full)
- 12. If TI = 1 (TX_BUF is Empty) then Send data to SBUF.**
13. Clear TI
14. Repeat step 2 to 13

Example #7...

➤ Program.

```
MOV SCON, #50H      ;8-bit, 1 stop, REN enabled
MOV TMOD, #20H     ;timer 1, mode 2(auto reload)
MOV TH1, #-3       ;9600 baud rate
SETB TR1           ;start timer 1
AGAIN: MOV A, P0    ; READ DATA FROM P0
L1:    JNB TI, L1   ;wait for the Tx_Buff = Empty , TI =1
      MOV SBUF, A   ;Send data serially
      CLR TI        ;clear TI for next char
      MOV A, P1     ; READ DATA FROM P1
L2:    JNB TI, L2   ;wait for the Tx_Buff = Empty , TI =1
      MOV SBUF, A   ;Send data serially
      CLR TI        ;clear TI for next char
      MOV A, P2     ; READ DATA FROM P2
L3:    JNB TI, L3   ;wait for the Tx_Buff = Empty , TI =1
      MOV SBUF, A   ;Send data serially
      CLR TI        ;clear TI for next char
      SJMP AGAIN    ;Repeat again
```

Thank You!!