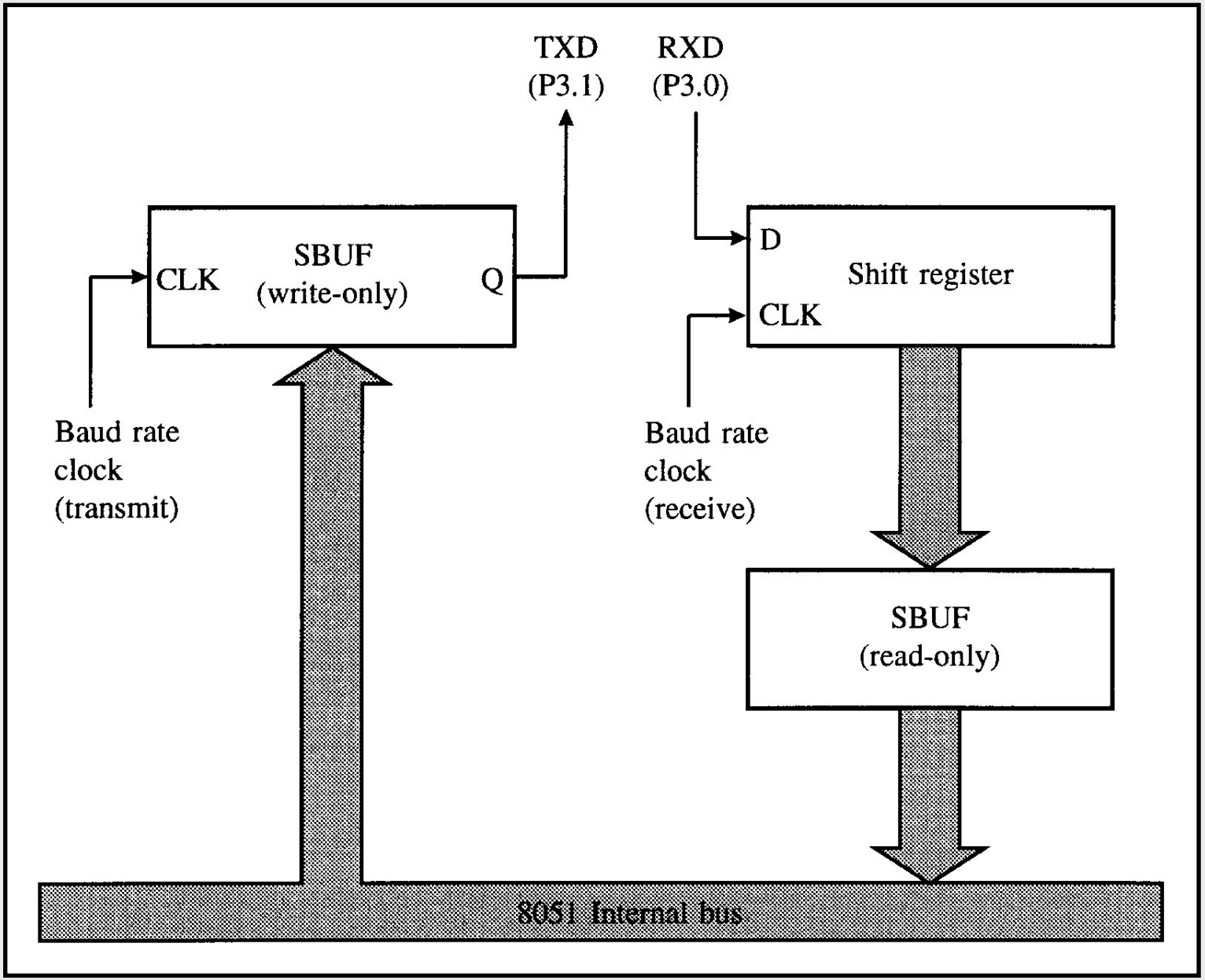
8051 serial port operation

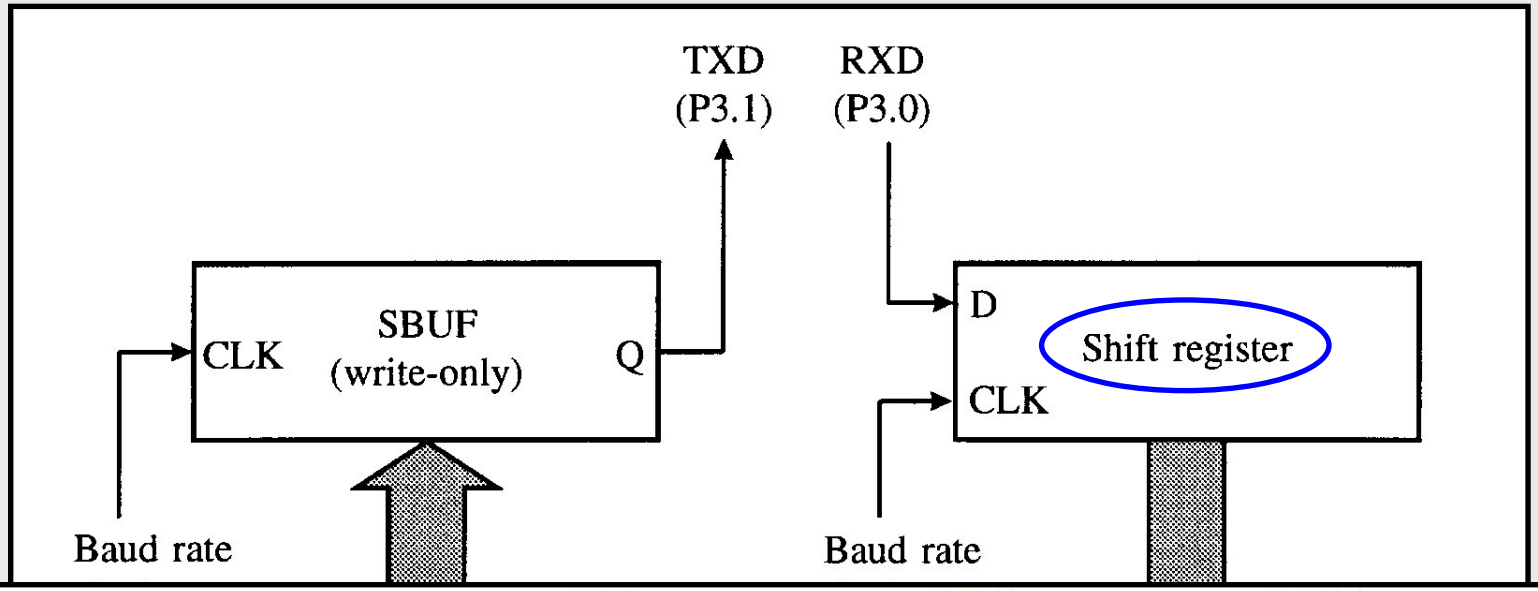
introduction

- output data : parallel-to-serial conversion
- input data : serial-to-parallel conversion
- hardware access by TXD, and RXD bits
- full-duplex operation
- receive buffering
- software access by SBUF, and SCON
- frequency of operation can be fixed (on-chip oscillator) or variable (Timer 1 should be programmed accordingly)

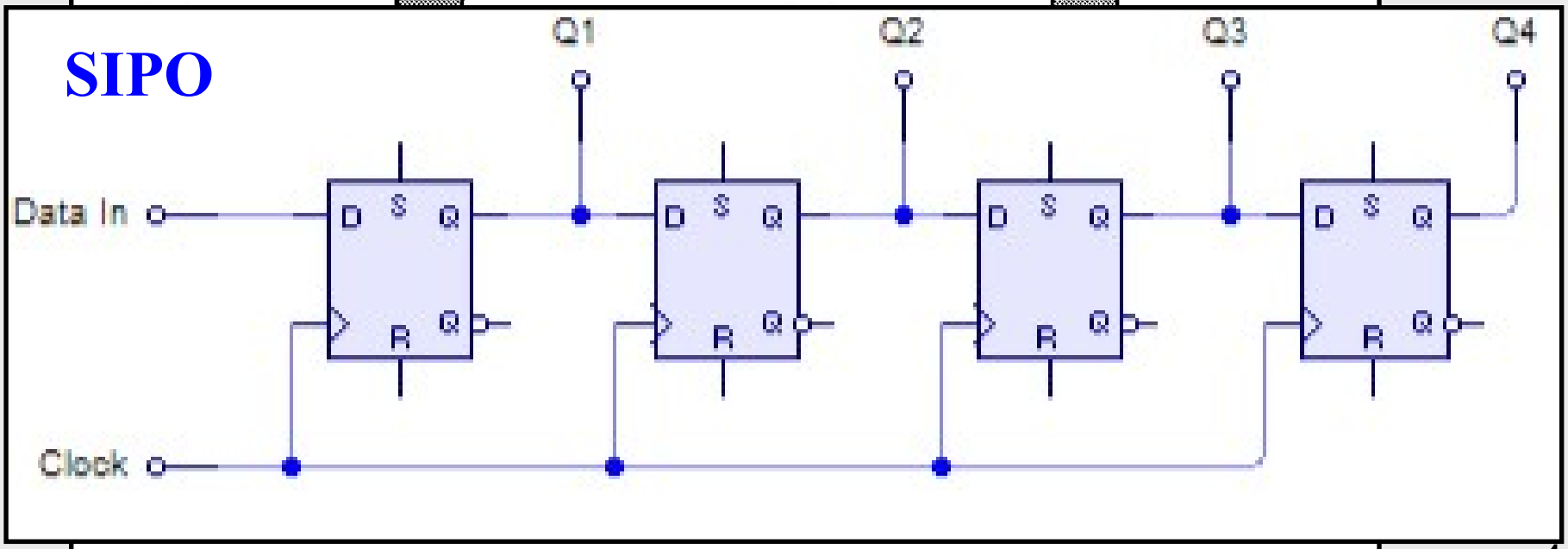
serial port block diagram



serial port block diagram



SIPO



Serial port CONtrol register

SCON (serial port control) register summary

BIT	SYMBOL	ADDRESS	DESCRIPTION
SCON.7	SM0	9FH	Serial port mode bit 0 (see Table 5–2)
SCON.6	SM1	9EH	Serial port mode bit 1 (see Table 5–2)
SCON.5	SM2	9DH	Serial port mode bit 2. Enables multiprocessor communications in modes 2 & 3; RI will not be activated if received 9th bit is 0
SCON.4	REN	9CH	Receiver enable. Must be set to receive characters
SCON.3	TB8	9BH	Transmit bit 8. 9th bit transmitted in modes 2 and 3; set/cleared by software
SCON.2	RB8	9AH	Receive bit 8. 9th bit received
SCON.1	TI	99H	Transmit interrupt flag. Set at end of character transmission; cleared by software
SCON.0	RI	98H	Receive interrupt flag. Set at end of character reception; cleared by software

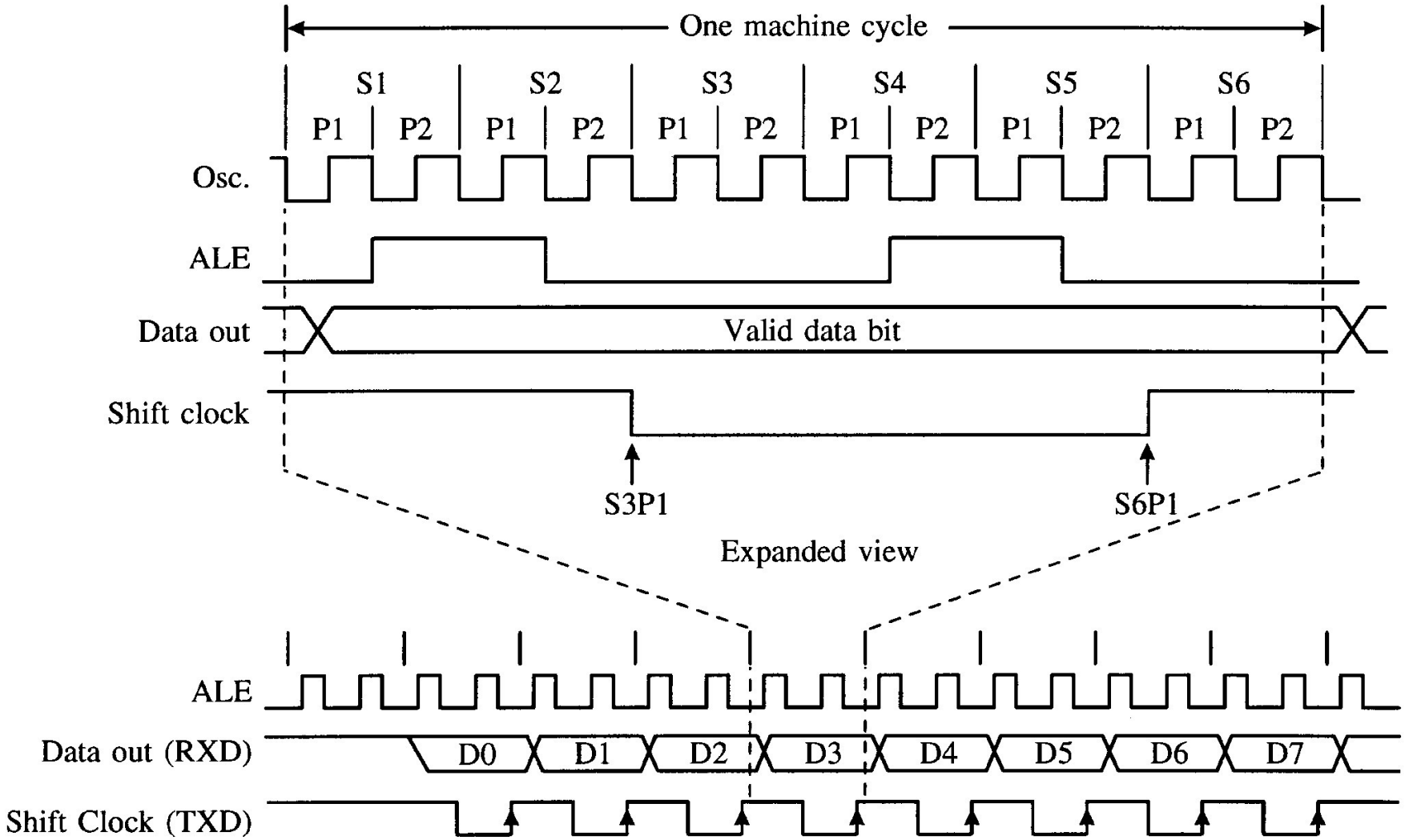
Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

8-bit shift register (mode 0)

- serial data enter and exit through RXD
- TXD outputs the shift clock
- 8-bit transmission (LSB first)
- baud rate is fixed at 1/12 oscillator frequency

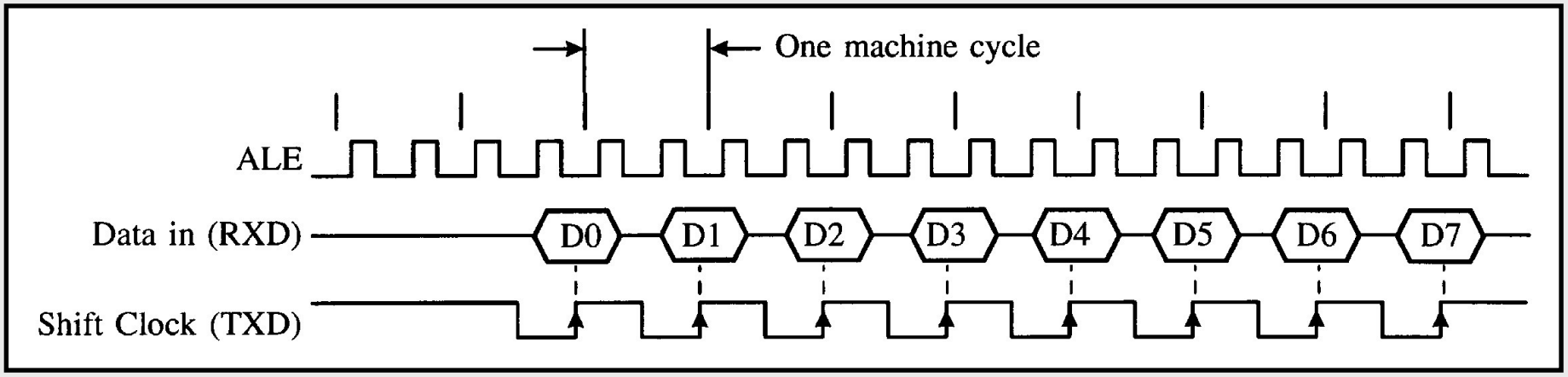
8-bit shift register (mode 0) (transmit)



initiated by any instruction that writes to SBUF

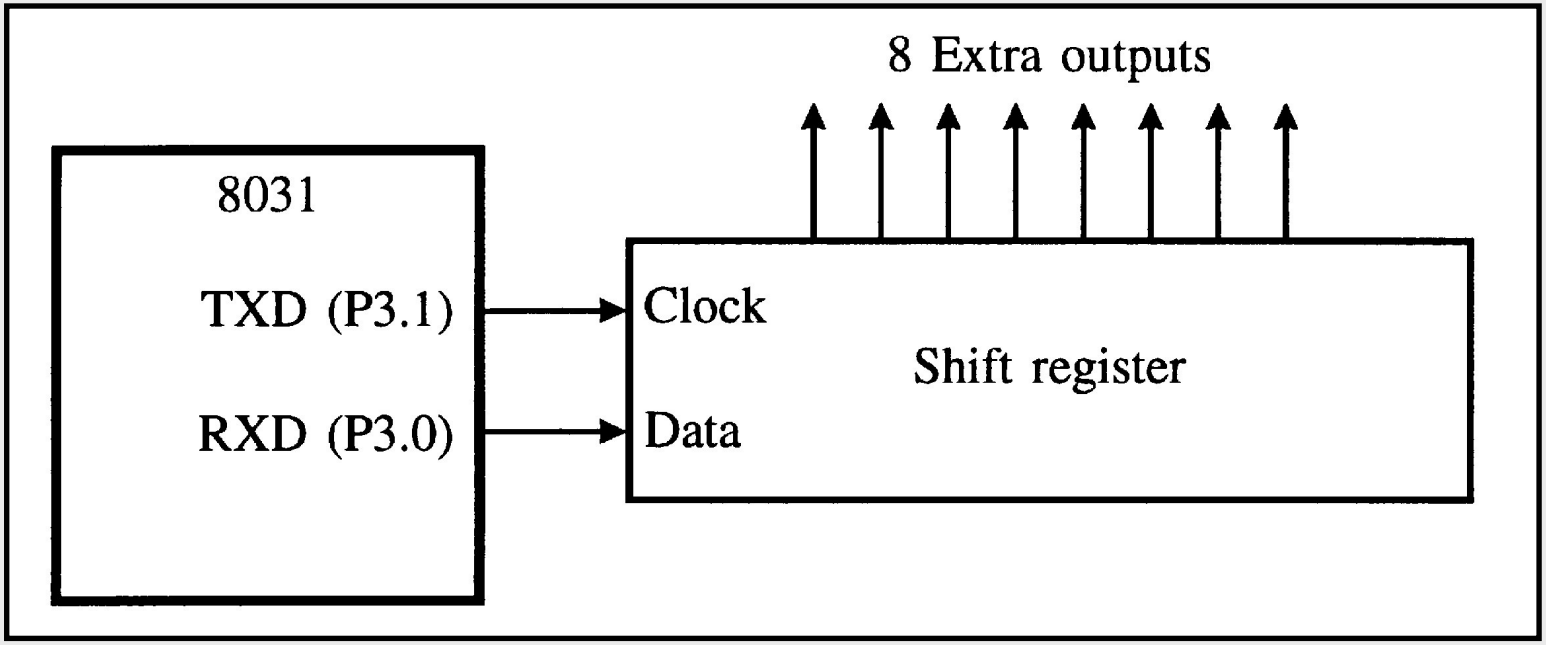
8-bit shift register (mode 0) (receive)

Receiption enabled : REN = 1, and RI = 0

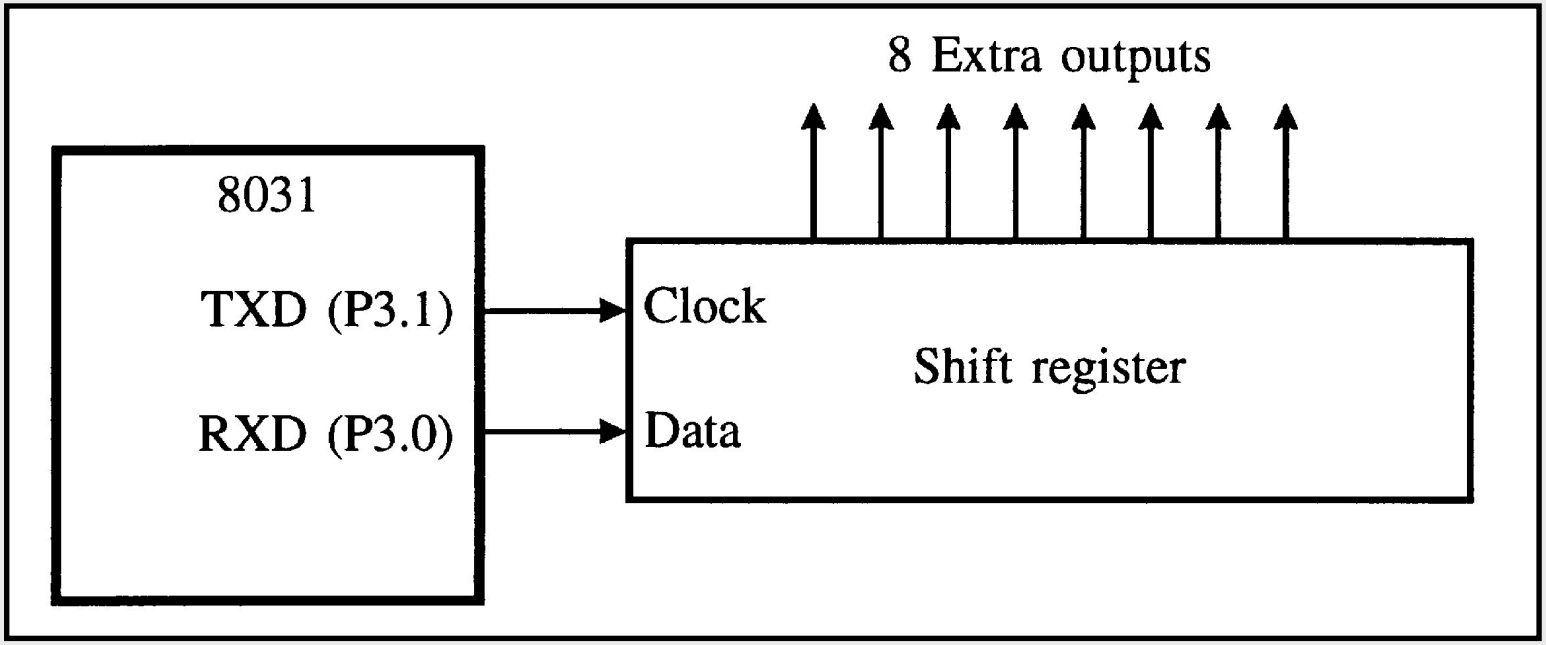


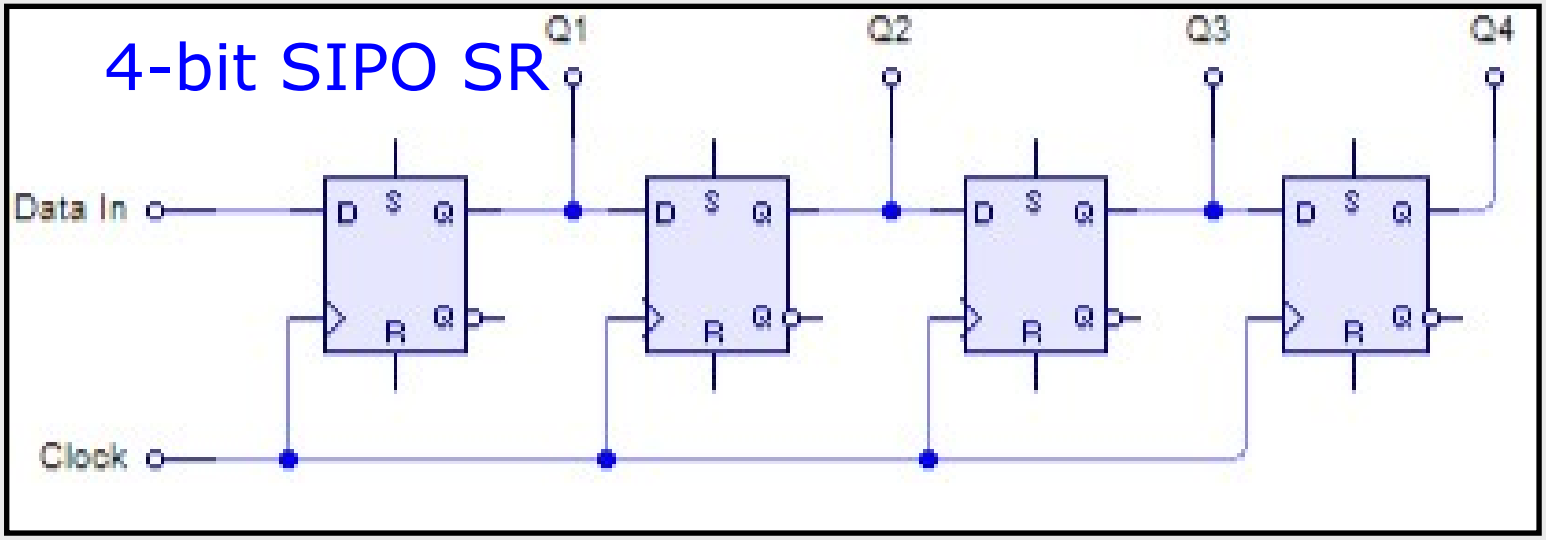
- set REN at the beginning of program (initialize)
- clear RI (begin data input operation)
- clock pulses on TXD, data clocked in RXD

expanding output capability of 8051



expanding output capability of 8051

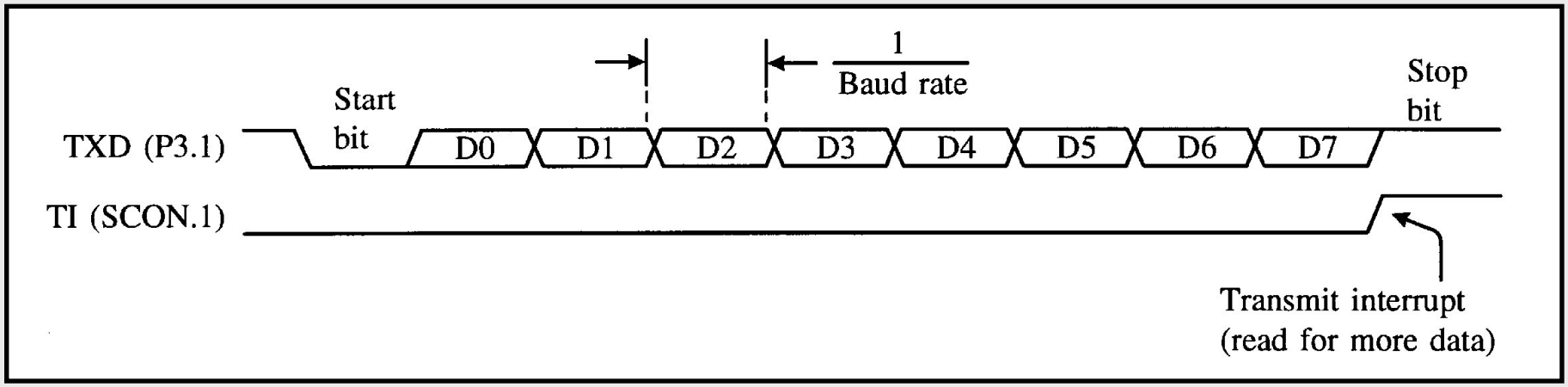




8-bit UART with variable baud rate (mode 1)

- Universal Asynchronous Receiver/Transmitter
- each character has start bit (low), stop bit (high)
- parity bit may be used between last bit & stop bit
- parallel-to-serial conversion for output data
- serial-to-parallel conversion for input data
- 10 bits are transmitted on TXD
- 10 bits are received on RXD
- stop bit goes into RB8 in SCON (receive mode)

8-bit UART with variable baud rate (mode 1)



- transmission is initiated by writing to SBUF
- TI is set with stop bit on TXD
- reception is initiated with 1-to-0 transmission on RXD

8-bit UART with variable baud rate (mode 1)

- false start bit detection on the receiver side by requiring a stop bit after eight counts following a 1-to-0 transition
- else receiver is assumed to be triggered by noise and returns to idle state
- if no problem, stop bit is clocked into RB8
- SBUF gets loaded with 8 data bits
- RI is set

- when RI = 0 AND
- SM2 = 1 & stop bit = 1 OR SM2 = 0

Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

8-bit UART with variable baud rate (mode 1)

- false start bit detection on the receiver side by requiring a stop bit after eight counts following a 1-to-0 transition
- else receiver is assumed to be triggered by noise and returns to idle state
- if no problem, stop bit is clocked into RB8
- SBUF gets loaded with 8 data bits
- RI is set

- when RI = 0 AND
- SM2 = 1 & 9th bit = 1 OR SM2 = 0

multiprocessor communication

Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ **32** or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

9-bit UART with fixed baud rate (mode 2)

- 9-bit UART with fixed baud rate
- 11 bits are transmitted and received
- 9^{th} data bit is programmable (e.g. parity bit)
- on transmission, 9^{th} bit is placed in TB8
- on receivement, 9^{th} bit is placed in RB8
- baud rate is either 1/32 or 1/64 of on-chip oscillator frequency

9-bit UART with variable baud rate (mode 3)

- similar to mode 2, but baud rate is programmable and provided by timer

initialization and accessing serial port registers

receiver enable

```
SETB    REN
```

```
MOV     SCON, #xxx1xxxxB
```

9th data bit

- loaded into TB8 by software for transmission
- received bit is written on RB8

even parity on 9th data bit

```
MOV     C, P
MOV     TB8, C
MOV     SBUF, A
```

odd parity on 9th data bit

```
MOV     C, P
CPL     C
MOV     TB8, C
MOV     SBUF, A
```

initialization and accessing serial port registers

even parity on 8th data bit

```
CLR     ACC.7       ; ensure MSB clear
MOV     C, P        ; copy to C
MOV     ACC.7, C    ; even parity into MSB
MOV     SBUF, A     ; send {7bits+even parity}
```

interrupt flags (RI, TI)

- set by hardware
- cleared by software

initialization and accessing serial port registers

even parity on 8th data bit

```
CLR     ACC.7      ; ensure MSB clear
MOV     C, P       ; copy to C
MOV     ACC.7, C   ; even parity into MSB
MOV     SBUF, A    ; send {7bits+even parity}
```

interrupt flags (RI, TI)

- set by hardware
- cleared by software

receive buffer full

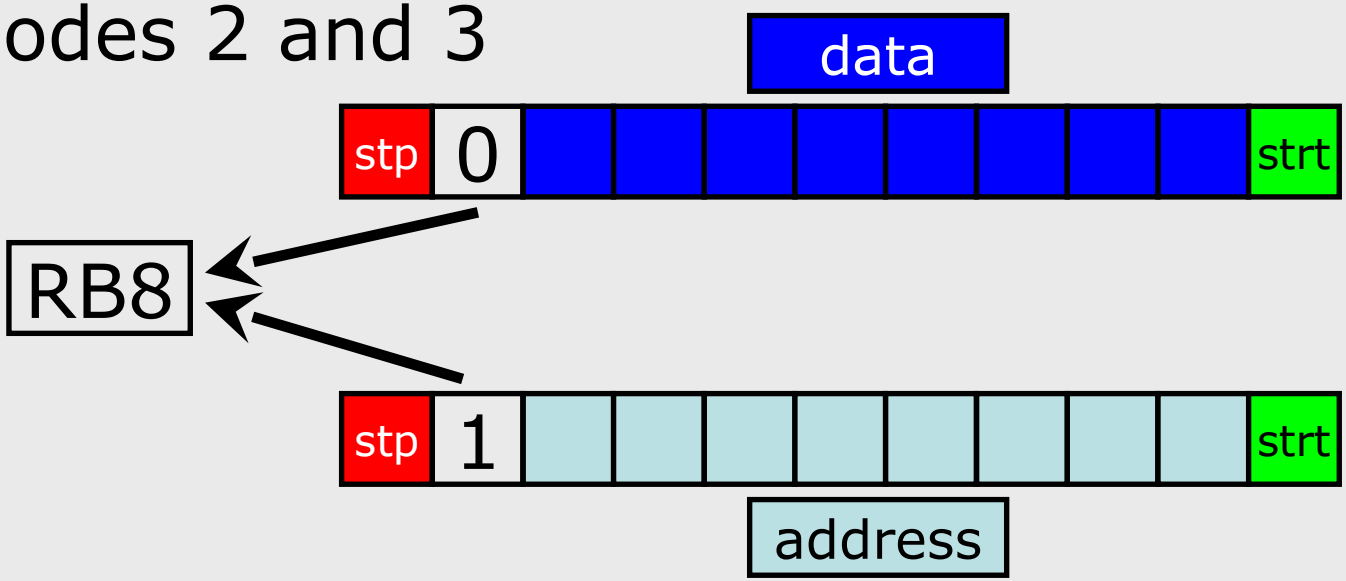
```
WAIT:  JNB   RI, WAIT
       CLR   RI
       MOV   A, SBUF
```

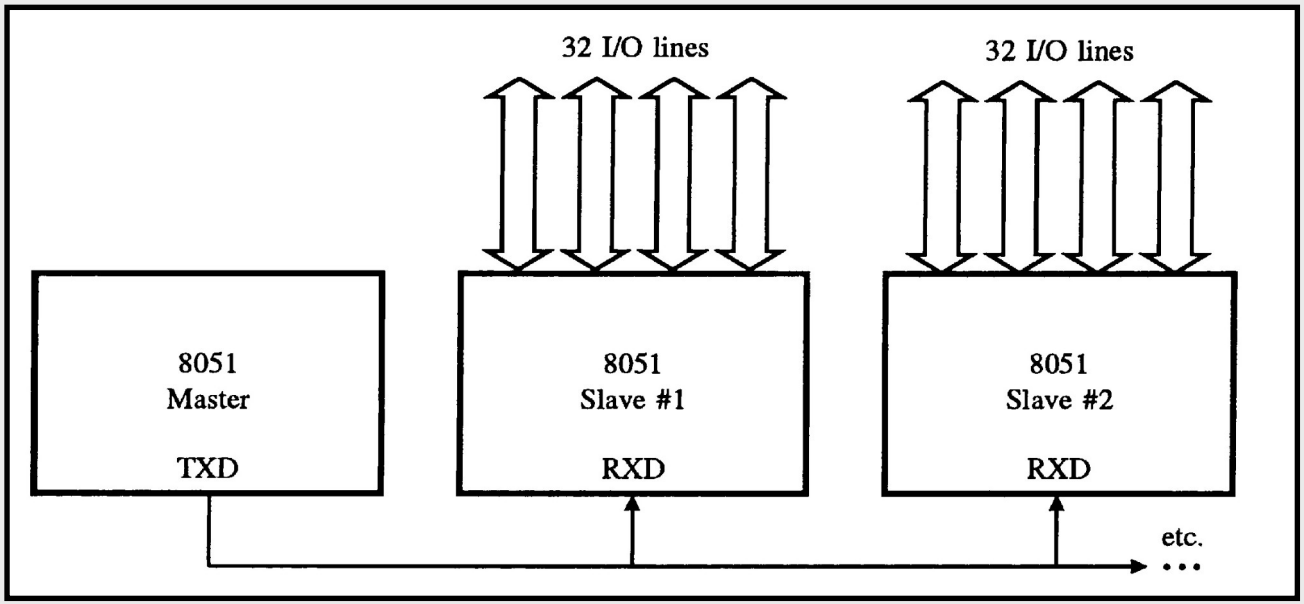
transmit buffer empty

```
WAIT:  JNB   TI, WAIT
       CLR   TI
       MOV   SBUF, A
```

multiprocessor communications

- for modes 2 and 3



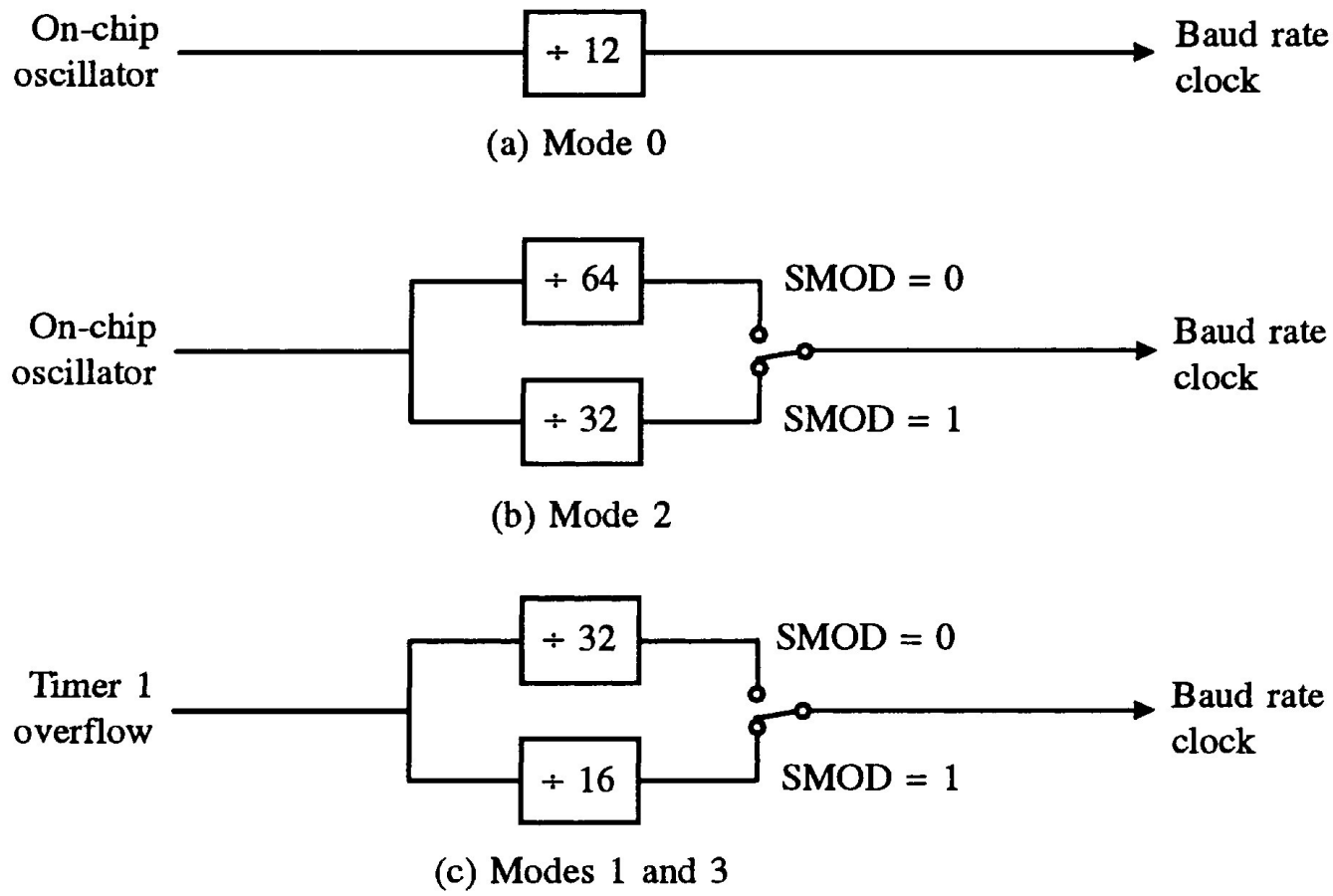


addressed slave clears its SM2 bit

serial port baud rates

Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)



bit 7 of PCON is SMOD bit

setting SMOD = 1 doubles the baud rate

SMOD is zero default by system reset

serial port baud rates (cont'd)

PCON is not bit-addressable

```
MOV     A, PCON     ; get current value
SETB    ACC.7       ; set bit 7 (SMOD)
MOV     PCON, A     ; write value back
```

- baud rates in modes 1 and 3 are determined by timer overflow rate

Baud rate summary

BAUD RATE	CRYSTAL FREQUENCY	SMOD	TH1 RELOAD VALUE	ACTUAL BAUD RATE	ERROR
9600	12.000 MHz	1	−7 (F9H)	8923	7%
2400	12.000 MHz	0	−13 (F3H)	2404	0.16%
1200	12.000 MHz	0	−26 (E6H)	1202	0.16%
19200	11.059 MHz	1	−3 (FDH)	19200	0
9600	11.059 MHz	0	−3 (FDH)	9600	0
2400	11.059 MHz	0	−12 (F4H)	2400	0
1200	11.059 MHz	0	−24 (E8H)	1200	0

5% error is tolerable due to *rounding*

initializing the serial port (example)

- 8-bit UART at 2400 baud (timer 1 to provide baud clock rate) (assume 12 MHz crystal oscillator)

initializing the serial port (example)

- 8-bit UART at 2400 baud (timer 1 to provide baud clock rate) (assume 12 MHz crystal oscillator)

Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

initializing the serial port (example)

- 8-bit UART at 2400 baud (timer 1 to provide baud clock rate) (assume 12 MHz crystal oscillator)

Serial port modes

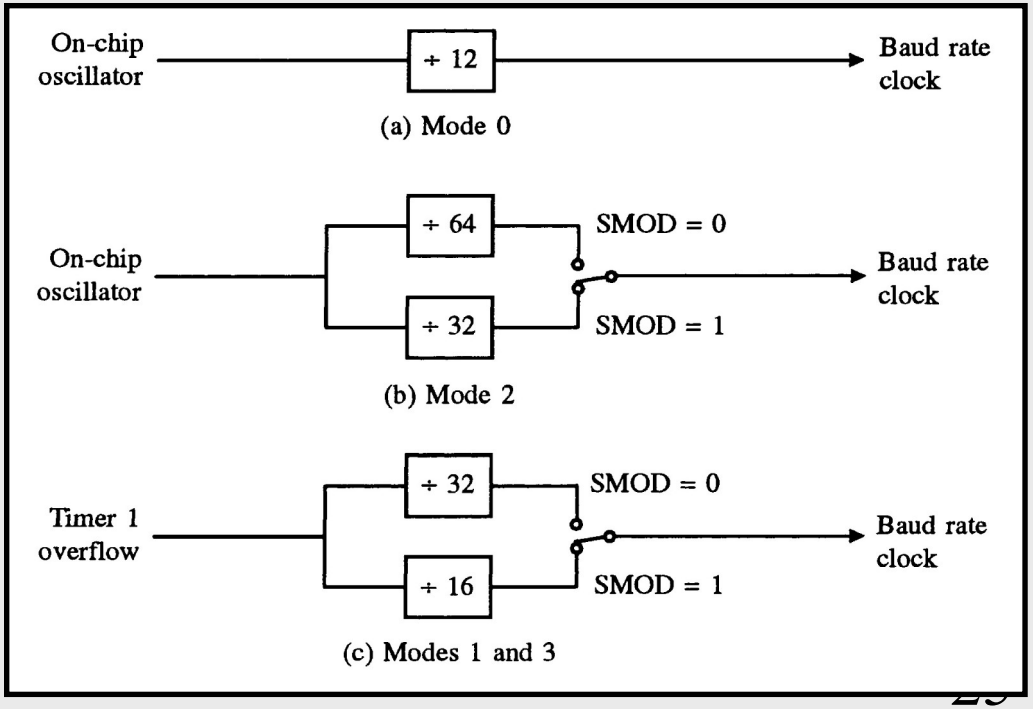
SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

initializing the serial port (example)

- 8-bit UART at 2400 baud (timer 1 to provide baud clock rate) (assume 12 MHz crystal oscillator)

Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

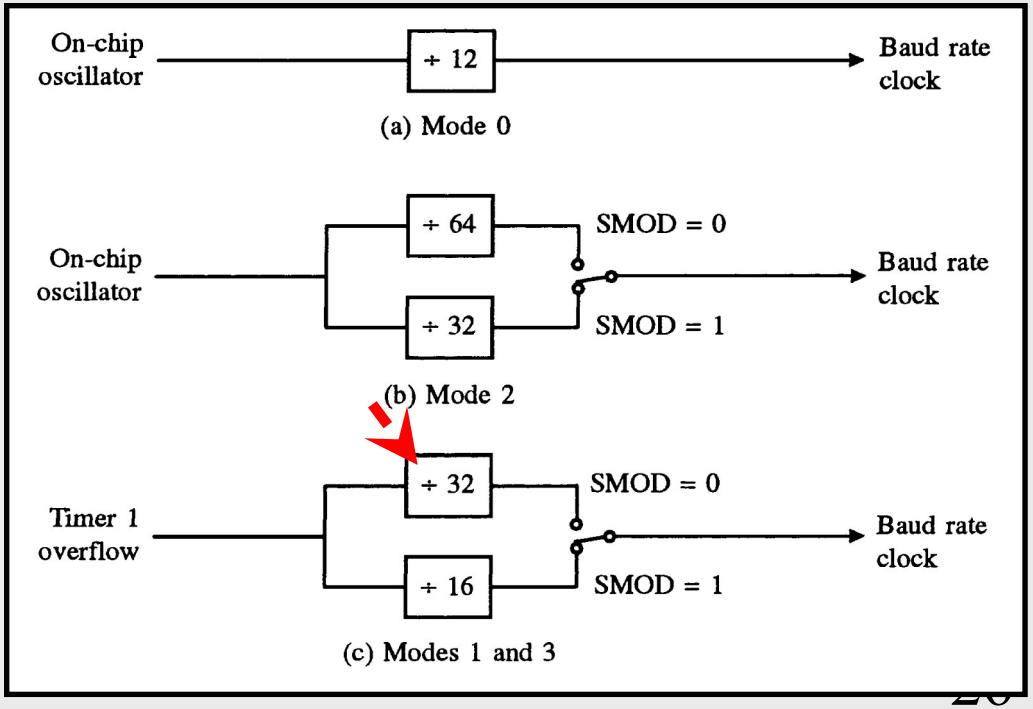


initializing the serial port (example)

- 8-bit UART at 2400 baud (timer 1 to provide baud clock rate) (assume 12 MHz crystal oscillator)

Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)



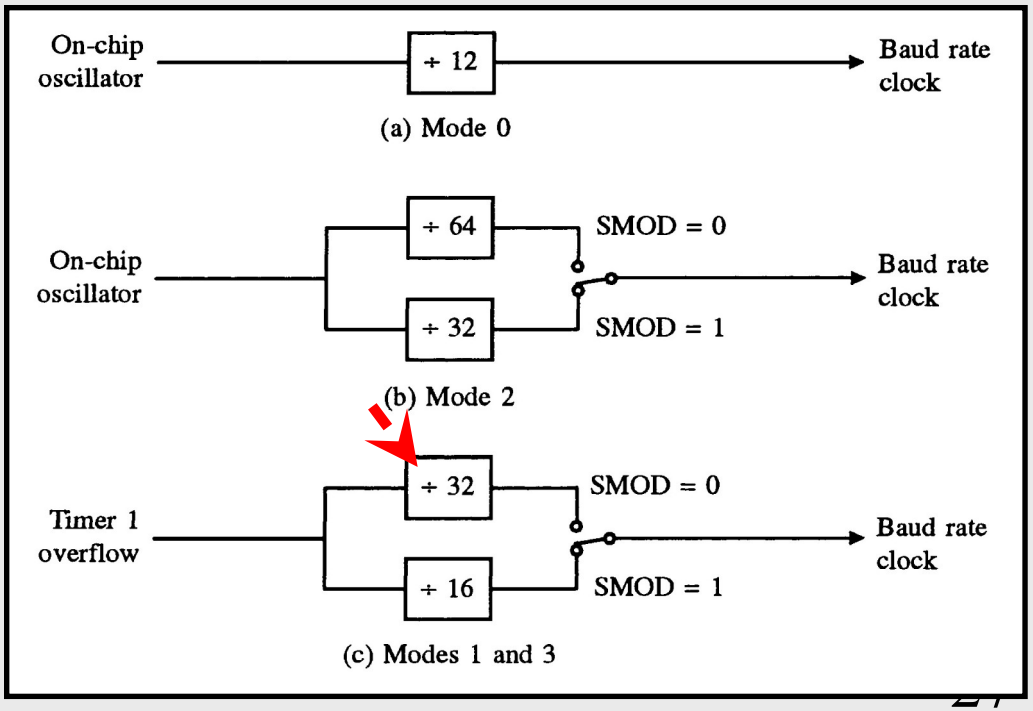
initializing the serial port (example)

- 8-bit UART at 2400 baud (timer 1 to provide baud clock rate) (assume 12 MHz crystal oscillator)

Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

2400 = T1OR / 32 *(SMOD = 0)*



initializing the serial port (example)

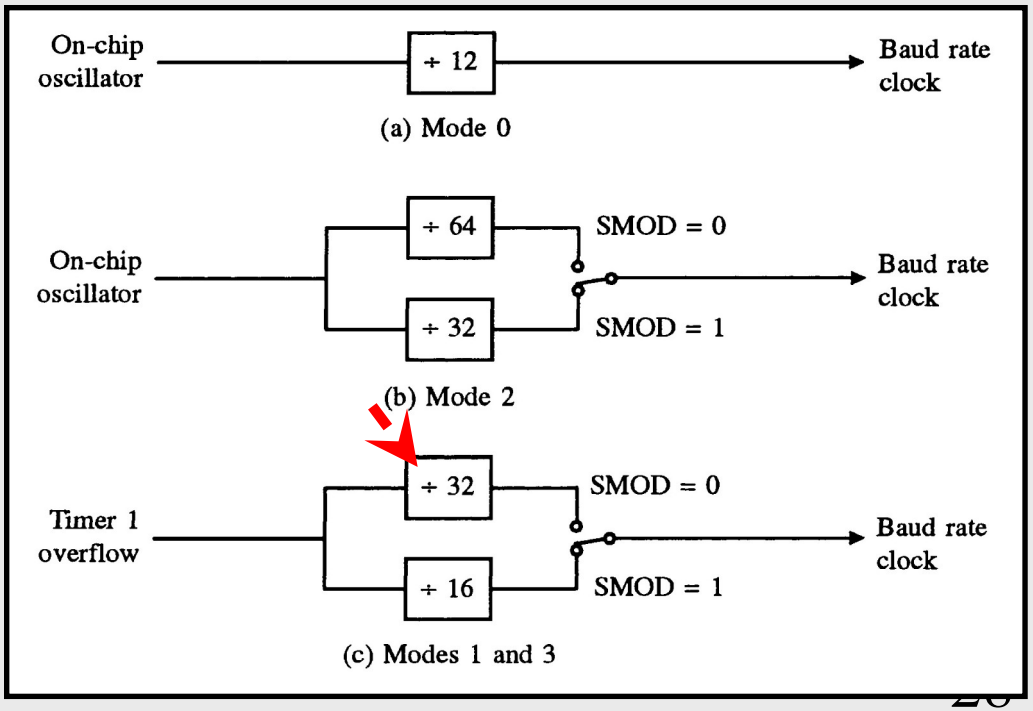
- 8-bit UART at 2400 baud (timer 1 to provide baud clock rate) (assume 12 MHz crystal oscillator)

Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

2400 = T1OR / 32 *(SMOD = 0)*

Timer 1 Overflow Rate = 76,8 kHz



initializing the serial port (example)

- 8-bit UART at 2400 baud (timer 1 to provide baud clock rate) (assume 12 MHz crystal oscillator)

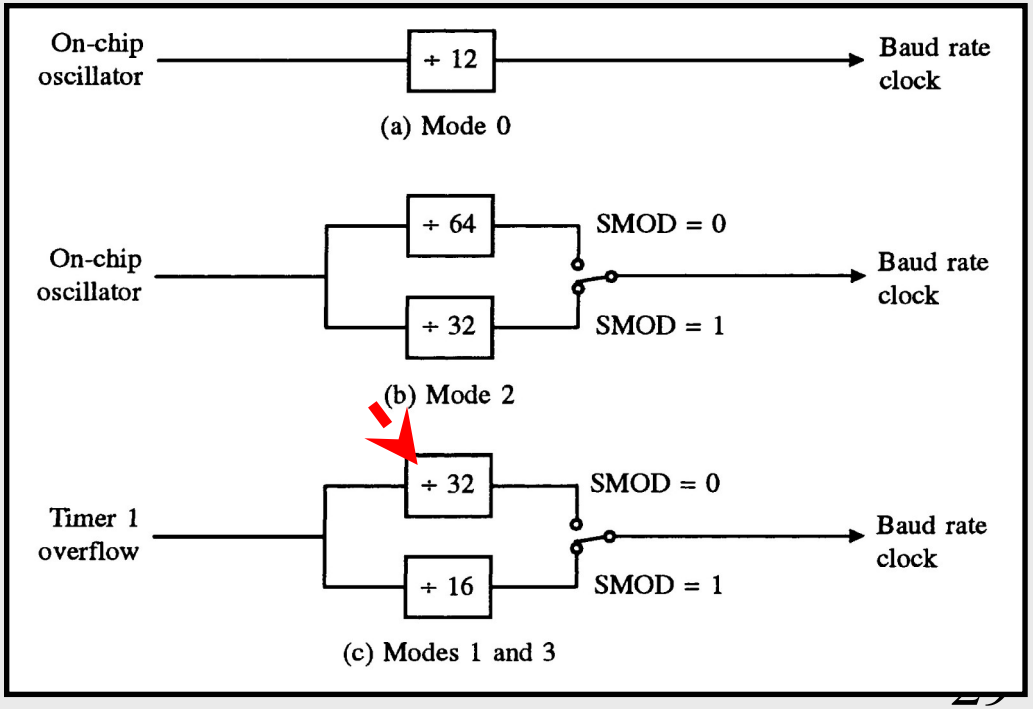
Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

2400 = T1OR / 32 *(SMOD = 0)*

Timer 1 Overflow Rate = 76,8 kHz

Timer 1 clocked at 1 MHz



initializing the serial port (example)

- 8-bit UART at 2400 baud (timer 1 to provide baud clock rate) (assume 12 MHz crystal oscillator)

Serial port modes

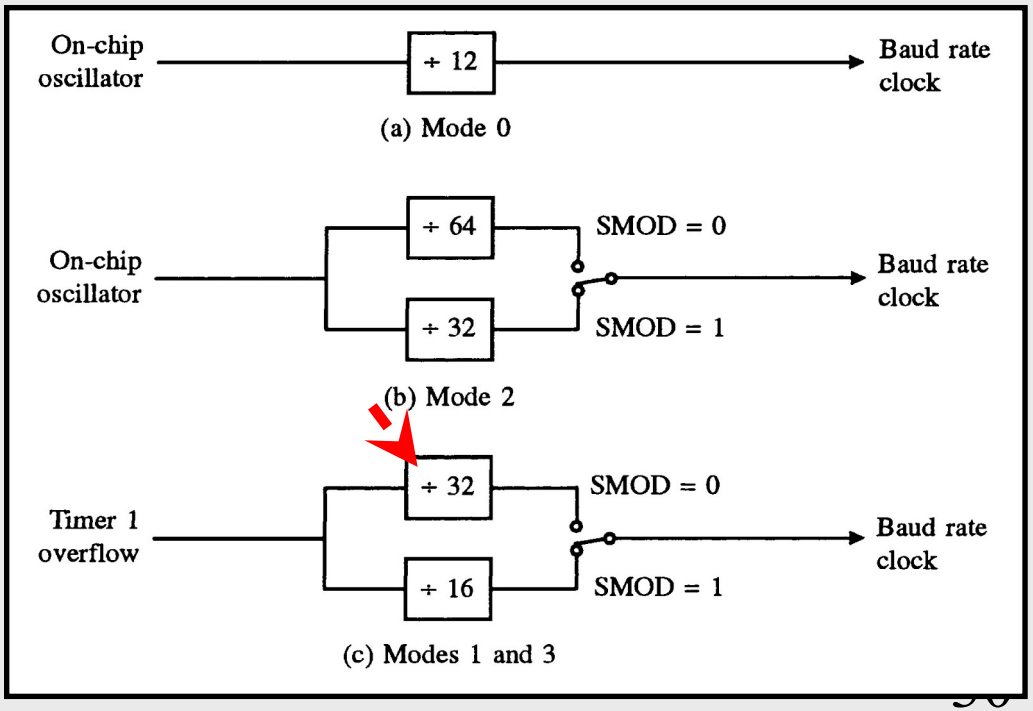
SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

2400 = T1OR / 32 *(SMOD = 0)*

Timer 1 Overflow Rate = 76,8 kHz

Timer 1 clocked at 1 MHz

# of reload = 1000000 / 76800



initializing the serial port (example)

- 8-bit UART at 2400 baud (timer 1 to provide baud clock rate) (assume 12 MHz crystal oscillator)

Serial port modes

SM0	SM1	MODE	DESCRIPTION	BAUD RATE
0	0	0	Shift register	Fixed (oscillator frequency ÷ 12)
0	1	1	8-bit UART	Variable (set by timer)
1	0	2	9-bit UART	Fixed (oscillator frequency ÷ 32 or ÷ 64)
1	1	3	9-bit UART	Variable (set by timer)

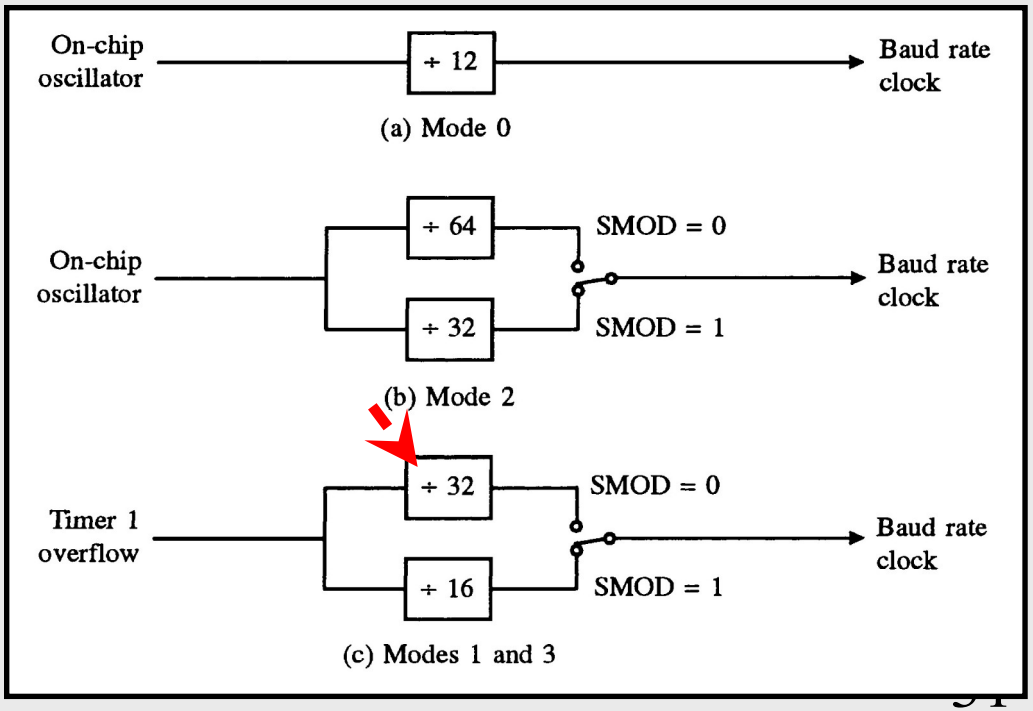
2400 = T1OR / 32 *(SMOD = 0)*

Timer 1 Overflow Rate = 76,8 kHz

Timer 1 clocked at 1 MHz

# of reload = 1000000 / 76800

# of reload = 13,02 ~ **13**



initializing the serial port (example)

- enable receive mode REN = 1
- indicate that transmit buffer is empty TI = 1
- set timer 1 to 8-bit autoreload mode (with proper TH1)

initializing the serial port (example)

- enable receive mode REN = 1
- indicate that transmit buffer is empty TI = 1
- set timer 1 to 8-bit autoreload mode (with proper TH1)

	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SCON:	0	1	0	1	0	0	1	0
	GTE	C/T	M1	M0	GTE	C/T	M1	M0
TMOD:	0	0	1	0	0	0	0	0
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TCON:	0	1	0	0	0	0	0	0
TH1:	1	1	1	1	0	0	1	1

initializing the serial port (example)

- enable receive mode REN = 1
- indicate that transmit buffer is empty TI = 1
- set timer 1 to 8-bit autoreload mode (with proper TH1)

	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SCON:	0	1	0	1	0	0	1	0
	GTE	C/T	M1	M0	GTE	C/T	M1	M0
TMOD:	0	0	1	0	0	0	0	0
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TCON:	0	1	0	0	0	0	0	0
TH1:	1	1	1	1	0	0	1	1

```
INIT:   MOV     SCON, #52H   ; serial port, mode 1
        MOV     TMOD, #20H   ; timer 1, mode 2
        MOV     TH1, #-13    ; reload for 2400 baud
        SETB    TR1          ; start timer 1
```

output character subroutine (example)

- OUTCHR subroutine that transmits 7-bit ASCII code on accumulator with 8th bit as odd-parity (return from subroutine by keeping the accumulator value intact)

output character subroutine (example)

- OUTCHR subroutine that transmits 7-bit ASCII code on accumulator with 8th bit as odd-parity (return from subroutine by keeping the accumulator value intact)

```
OUTCHR: MOV   C, P        ; parity in C flag
        CPL   C           ; odd-parity
        MOV   ACC.7, C    ; add to character code
AGAIN:  JNB   TI, AGAIN   ; empty check
        CLR   TI          ; TI rdy, clear TI flag
        MOV   SBUF, A     ; send character
        CLR   ACC.7       ; strip-off parity
        RET               ; return from subr.
```

output character subroutine (example)

- OUTCHR subroutine that transmits 7-bit ASCII code on accumulator with 8th bit as odd-parity (return from subroutine by keeping the accumulator value intact)

```
OUTCHR: MOV   C, P        ; parity in C flag
        CPL   C           ; odd-parity
        MOV   ACC.7, C    ; add to character code
AGAIN:  JNB   TI, AGAIN   ; empty check
        CLR   TI          ; TI rdy, clear TI flag
        MOV   SBUF, A     ; send character
        CLR   ACC.7       ; strip-off parity
        RET               ; return from subr.
```

```
MOV     A, # 'Z'
CALL    OUTCHR
(continue)
```

more general version

input character subroutine (example)

- INCHR subroutine that receives 7-bit ASCII character and expects 8th bit as odd-parity to set the carry flag if there is an error on parity

input character subroutine (example)

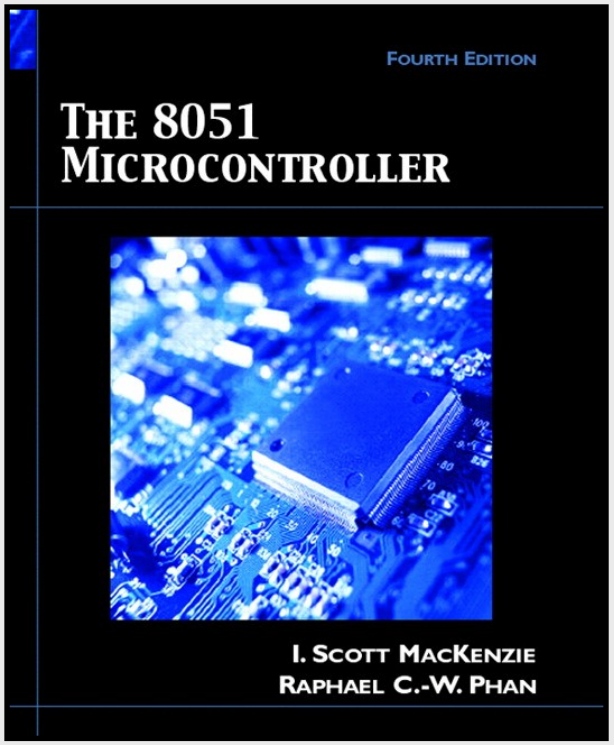
- INCHR subroutine that receives 7-bit ASCII character and expects 8th bit as odd-parity to set the carry flag if there is an error on parity

```
INCHR:  JNB     RI, $       ; wait for character
        CLR     RI          ; clear flag
        MOV     A, SBUF     ; read character into A
        MOV     C, P        ; P set for odd-parity
        CPL     C           ; check for error
        CLR     ACC.7       ; strip-off parity
        RET
```

summary

- serial port operation modes
- baud rate calculation for serial operation
- setting up serial port for UART

references



Google™

WIKIPEDIA
The Free Encyclopedia