8051 Timer/Counter

Two internal Timers/Counters
- 16-bit timer/counter
- Timer uses system clock as source of input pulses
- Counter uses external input pulses from port 3 (T0,T1)
- If associated interrupt is enabled, when count overflow an interrupt is generated

Registers
- TH0, TL0 : timer/counter register of timer 0
- TH1, TL1 : timer/counter register of timer 1
- TMOD : Mode Select register
- TCON : Control Register
### 8051 Timer/Counter

<table>
<thead>
<tr>
<th>Timer/Counter 0</th>
<th>Timer/Counter 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TMOD</strong></td>
<td><strong>TCON</strong></td>
</tr>
<tr>
<td>M0</td>
<td>IT0</td>
</tr>
<tr>
<td>M1</td>
<td>IE0</td>
</tr>
<tr>
<td>C/T#</td>
<td>IT1</td>
</tr>
<tr>
<td>GATE#</td>
<td>IE1</td>
</tr>
<tr>
<td></td>
<td>TR0</td>
</tr>
<tr>
<td></td>
<td>TF0</td>
</tr>
<tr>
<td></td>
<td>TR1</td>
</tr>
<tr>
<td></td>
<td>TF1</td>
</tr>
</tbody>
</table>
8051 Timer/Counter

Operation Modes

Mode 0
- 13-bit counter, an interrupt is generated when counter overflows
- It takes 8192 input pulses to generate the next interrupt

Mode 1
- 16-bit counter, similar to mode 0, but take 65536 input pulses
8051 Timer/Counter

Operation Modes (cont.)

Mode 2
- 8-bit reload
- TLi operates as timer/counter
- THi store a number and reload to TLi when overflows

Mode 3
- Timer 1 is inactive, hold count value
- TL0 and TH0 operate as two separate 8-bit timer/counter
- TL0 control by timer 0 control bits
- TH0 operate as timer driven by system clock, prescaled by 12 and cause timer 1 interrupt overflows
**8051 Timer/Counter**

Mode 0, like mode 1, except that it’s 13-bit timer/counter
Setup the timer 0, mode 1 -- 16-bit timer

```assembly
timer_init:
    mov TMOD, #1
    setb TR0
    mov TH0, #H_count
    mov TL0, #L_count
    ret

In ISR, count value must be restored in subroutine
```

```assembly
timer0_isr:
    mov TH0, #H_count
    mov TL0, #L_count
    call something
    reti
```
8051 Timer/Counter

Setup the timer 0, mode 2 -- 8-bit timer auto-reload

timer_init:
  mov TMOD, #2
  setb TR0
  mov TH0, #count
  ret

Mode 2, value is reload automatically from THx

timer0_isr:
  call something
  reti
Setup the timer, mode 3

timer_init:
  mov TMOD, #3
  mov TH0, #Timer1_count
  mov TL0, #Timer0_count
  setb TR0
  setb TR1
  ret

timer0_isr:
  mov TL0, #Timer0_count
  call something0
  reti
8051 Timer/Counter

Timer mode 3

Mode 3, likes mode 1, but it is:

- 8-bit counter
- TL0 is used for timer0
- TH0 is used for timer1
- TH1 & TL1 can be used for serial or other application
8051 Serial Port

SCON Register (cont.)

- SM1 & SM0 Serial Mode (bit 6 & 7)
  - Operating modes

Operating modes

- Mode 0 8-bit shift register, f/12
  - 1Mbit with 12Mhz Oscillator Frequency
- Mode 1 8-bit UART, variable baud rate
- Mode 2 9-bit UART, f/64 or f/32
  - 187.5K and 375K with 12MHz Oscillator Frequency
- Mode 3 9-bit UART, variable baud rate
### 8051 Serial Port

#### Variable Baud Rate

Baud Rate = \( \frac{2 \times f}{384 \times (256 - TH1)} \)

<table>
<thead>
<tr>
<th>Baud Rate</th>
<th>f</th>
<th>SMOD</th>
<th>TH1</th>
</tr>
</thead>
<tbody>
<tr>
<td>62.5K</td>
<td>12.000</td>
<td>1</td>
<td>FFh</td>
</tr>
<tr>
<td>19.2K</td>
<td>11.059</td>
<td>1</td>
<td>FDh</td>
</tr>
<tr>
<td>9.6K</td>
<td>11.059</td>
<td>0</td>
<td>FDh</td>
</tr>
<tr>
<td>4.8K</td>
<td>11.059</td>
<td>0</td>
<td>FAh</td>
</tr>
<tr>
<td>2.4K</td>
<td>11.059</td>
<td>0</td>
<td>F4h</td>
</tr>
<tr>
<td>1.2K</td>
<td>11.059</td>
<td>0</td>
<td>E8h</td>
</tr>
</tbody>
</table>
8051 Serial Port

Initialize Serial Port Subroutine

Set baud rate to 9600 for 11.0592 Crystal

```assembly
    mov     TMOD, #20h ; set timer 1 for auto reload
    mov     TCON, #41h ; run counter 1 edge trig
    mov     TH1, #0FDh ; 9600 baud
    mov     SCON, #50h ; 8-bit data mode 1
```

Sending and Receiving Subroutines

```assembly
    sendchr:                           getchr:
        clr    TI                         jnb    RI, getchr
        mov    SBUF, A                mov    A, SBUF
    txloop:      jnb    TI, txloop     clr     RI
    ret                ret
```