## 1.5 Timers and Serial port:

### 1.5.1 Timer/Counters

- The 8051 has two 16-bit timer/counters. These two timer/counters can be programmed independently.
- There is a bit in the TMOD SFR that specifies whether it is a timer or a counter.
- If this bit is set, the timer/counter will work as a counter; and
- if this bit is reset, the timer/counter will work as a timer.

**Timer Mode**

- When a timer/counter is functioning as a timer, the timer register (TH 1 and/or TL1 for Timer 1 or TH0 and/or TL0 for Timer 0) is incremented after every machine cycle.
- That is, it will be working at 1/12th of the oscillator frequency because each machine cycle has got 12 oscillator periods.
- As an example, after starting the spin motor in a washing machine, the next operation, i.e. the motor shut down, is performed after a fixed time interval.

Modes of Timer:

These timers can function in four different modes, namely:

**i) mode 0, ii) mode 1,iii) mode 2 and iv) mode 3.**

- These modes are achieved by setting certain bits in the TMOD-register.
- Mode 0 to mode 2 are common for both the timers but not mode 3

**i) Mode 0:**

- In this mode, TL0and TH0 for Timer 0 (or TLl and THl for Timer 1) are used as a 13- bit register,
- i.e. all the 8 bits of the TL0 or TLl are utilized and the five lower most bits of the TH0 or THl are used for counting purposes.
- As the count rolls over from all 1s in the register to all 0s, the interrupt flag is set.

- This timer interrupt flag is a bit, namely TF0 (for Timer 0) in the TCON, which is a special function register.
- From Figure 1, it is clear that if C/T = 0, then the register is incremented after every machine cycle.

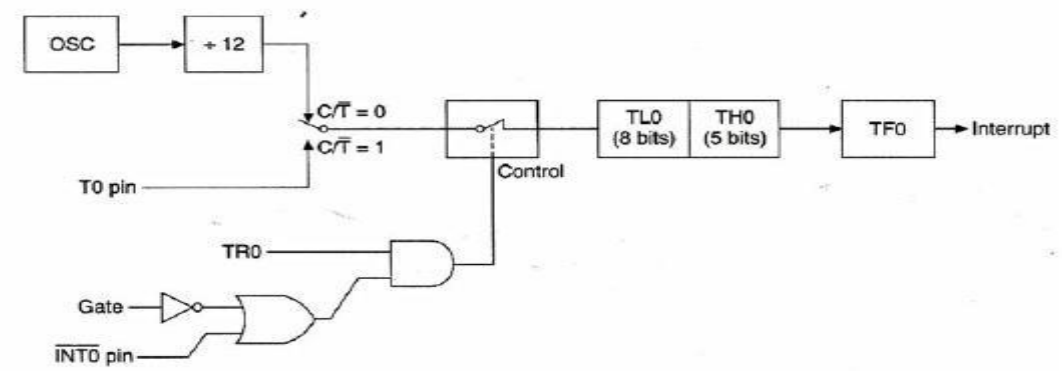



**Figure 1: Timer0, mode0 _13-bit Counter**

- **Case 1:**

TR0 = 1 (high)

Gate = 0 (low) and INT0= 0 (low)

- **Case 2 :**

TRO = 1 (high)

INTO = 1 (high) Gate = 1 (high)

- Mode 0 for both the timer/counters is the same, the only thing to do is to change TR0 to TR1, INT0' to INTl', and Gate bit for Timer 0 to Gate bit for Timer 1.

**ii)Mode 1:**

This mode (Figure 2) is similar to mode 0, except that in this, 16 bits, that is, full TL0 and

TH0 for Timer 0 (or TLl andTH1 for Timer 1) are used for counting. So, the interrupt flag will be

set only when the 16 bits go from all 1s to all 0s.

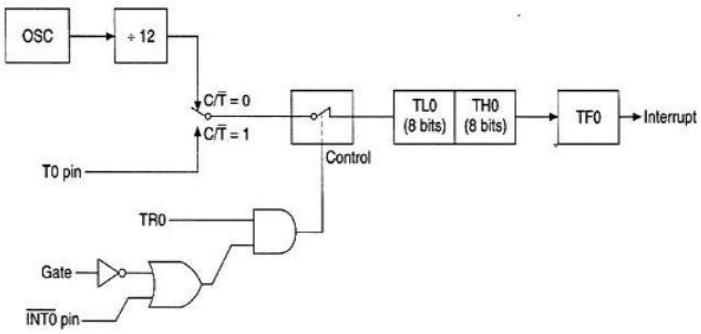


**Figure 2: Timer 0, mode 1 _ 16-bit counter.**

**iii) Mode 2:**

- In this mode, the timer register is 8 bits wide. TL0 for Timer 0 (or TLl for Timer 1) is used for this purpose (Figure 3).
- This mode is also called the auto-reload mode as the timer generates an interrupt on overflow and after generating the interrupt, will also reload the present value from TH0 into TL0. This present value can be put in the TH0 through software.
- The interrupt flag is set when TL0 goes from all 1s to all 0s.After generating an interrupt, it also reloads the TL0 with the value from TH0 (80H in this case) and then starts counting again.

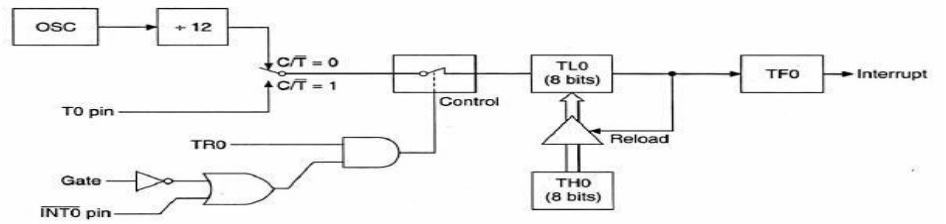



**Figure 3: Timer 0, mode 2 _ auto reload.**

**iv) Mode 3:**

- "If the Timer 0 is put into mode 3 (Figure 4), then it acts as two 8-bit counters(TL0 and TH0 become two separate counters).
- In this case, all the Timer 0 control bits (C / T',Gate, TRO, TFO and INT0') are used by TL0 itself and TH0 register is locked into a timerfunction.
- TH0 is counting machine cycles and has taken over the use of TRl and TFl from Timer 1.
- Therefore TH0 will now control Timer 1 interrupt. If the Timer 1 is put into mode 3,it just holds the count.
- The effect is same as setting TRI = 0, hence opening the switch.

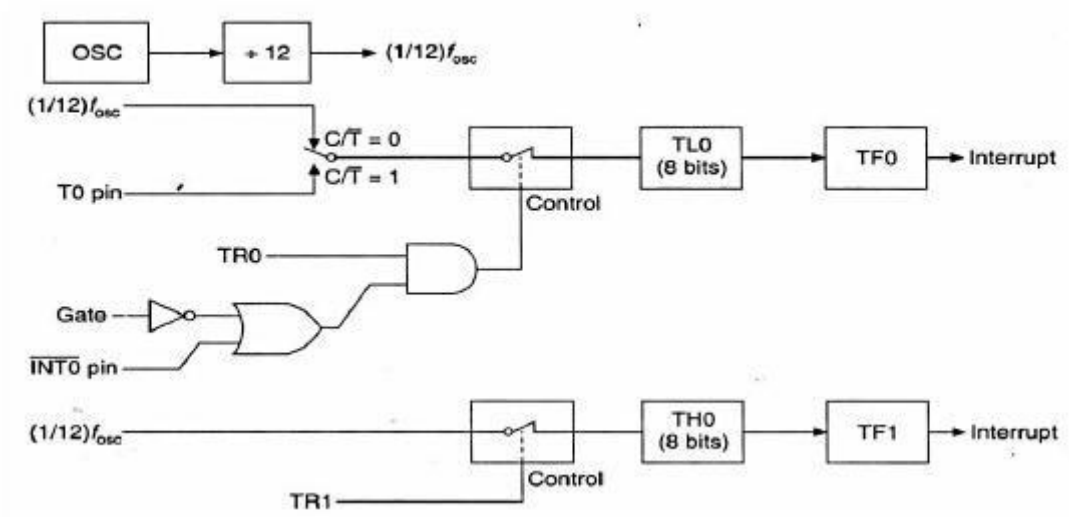



**Figure 4: Timer 0, mode 3 _ split to two 8-bit counters**

**Timer Control and Status Register:**

- The special function registers TMOD and TCON are used to control the timer/counterfunctions.
- When we write into these registers, the data is latched into them and takes effect atS1P1 of

the next instruction (first cycle).

The bits of these registers are described below:

1. **TMOD: Timer Mode Control Registers**

| | Timer 1 | | | | Timer 0 | | | |
|---|---|---|---|---|---|---|---|---|
| *Bit no.* | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| *Symbol* | Gate | C/$\overline{T}$ | M1 | M0 | Gate | C/$\overline{T}$ | M1 | M0 |

M1 and M0 specify the mode as follows:

| *M1* | *M0* | *Mode* | *Description in brief* |
|---|---|---|---|
| 0 | 0 | 0 | 13-bit counter |
| 0 | 1 | 1 | 16-bit counter |
| 1 | 0 | 2 | 8-bit counter with autoreload |
| 1 | 1 | 3 | Split Timer 0 into two 8-bit counters or to stop Timer 1 |

- If C/T = 1, the timers function as counters to count the negative transitions' at T0 or T1 pins,
- If C/T= 0, the timers function as timers, that is, they basically count the number of machine cycles.
- Gate = 0 means that the timer is controlled by TR1 or TR0 only, irrespective of INT0' or INT1'.
- Gate = 1 means that the timer control will depend on INT0'or INT1' and also on TR0 or TR1 bits.

## 2. TCON: Timer Control Register

| *Bit no.* | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| *Symbol* | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |

- **TF 1:**Timer 1 overflow flag. Set by hardware when the timer/counter overflows. Cleared by hardware when the processor vectors to the interrupt routine.
- **TR1:**Timer 1 run control bit. Set/cleared by software to turn the timer/counter on/off.
- **TF0:** Timer 0 overflow flag. Set by hardware when the timer/counter overflows. Cleared by hardware when the processor vectors to the interrupt routine.
- **TR0:** Timer 0 run control bit. Set/cleared by software to turn the timer/counter on/off.
- **IE 1:** Interrupt 1 edge flag. Set by hardware when the external interrupt edge is

Detected. Cleared when the interrupt is processed.

- **IT1:** Interrupt 1 type control bit. Set/cleared by software to specify the falling edge/low level triggered external interrupts. When IT1 = 1 then INTl' is falling edge triggered, otherwise when IT1 = 0 the INT1' is low-level triggered.

- **IE0:** Interrupt 0 edge flag. Set by hardware when the external interrupt edge is detected. Cleared when the interrupt is processed.

- **IT0**: Interrupt 0 type control bit. Set/cleared by software to specifiy the falling edge/low level triggered external interrupts. When IT0 = 1 then INT0' is falling edge triggered, otherwise when IT0 = 0 the INT0' is low-level triggered.

### 1.5.2 Serial Communication

- In serial communication, a single data can be used instead of 8-bit data lines.
- It is uses either asynchronous data format or synchronous data format methods.
- Serial transmission and reception can be easily achieved using UART devices.
- 8051 has built in UART, communication circuit use SUBF register to hold the data.
- Port 3, bit-0 act as RXD pin and bit-1 act as TXD for serial port.
- SUBF has two registers, one for write only TXD pin and another one for read only RXD pin.
- It can be programmed in 4 types of modes.
- The modes are: mode 0, mode 1, mode 2, and mode 3.

**Mode 0:**

- In serial communication are done through RXD pin.
- TXD pin acts as shift register clock and rising edge of the clock.
- Baud rate of the mode 0 is fixed to ½ of oscillator frequency

**Mode 1:**Standard UART

- It is a standard asynchronous serial communication mode.10-bits are transmitted or received, Baud rate is variable,

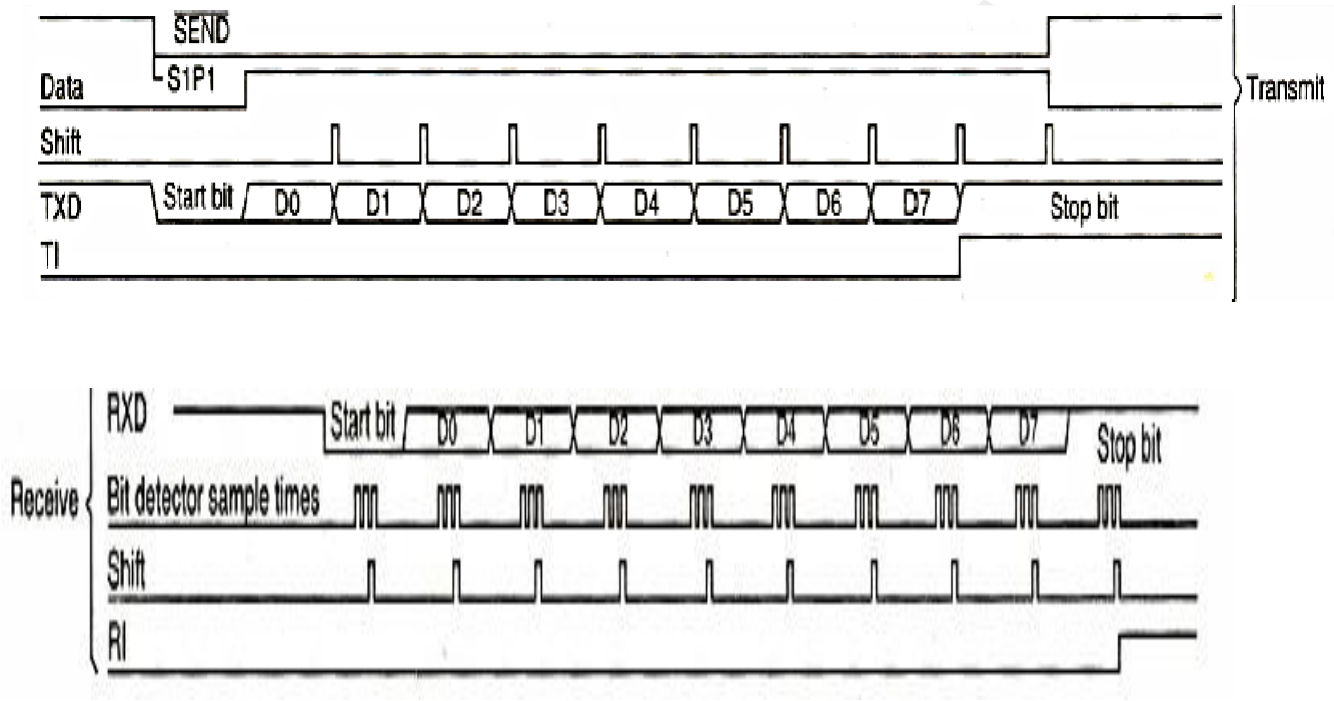


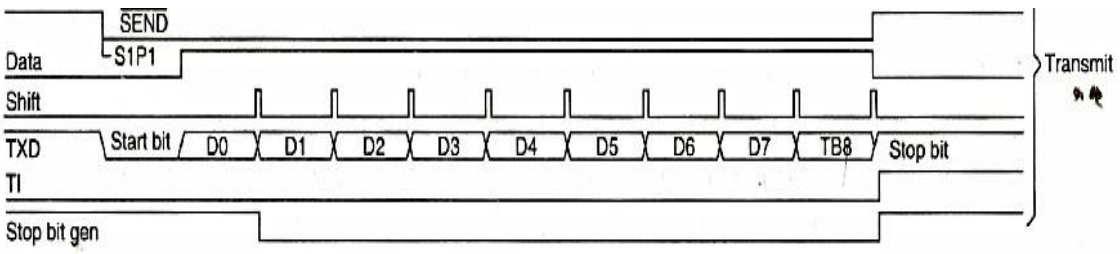

**Figure 6: mode 1**

**Mode2: Multiprocessor Mode**

- 11-bits are transmitted through TXD or received through RXD.
- it consists of one start bit(0),8 data bits, programmable 9thbit and a stop bit(1).
- On transmission 9th data bit is 0/1, 8-bit is loaded into SBUF and copied from TB8 in SCON register. On reception it goes into RB8 of SCON register of SFR.

**Mode 3:**

- it is same as mode 2 except baud rate, since mode 3 supports variable baud rate and calculated similar to mode1 using timer1.



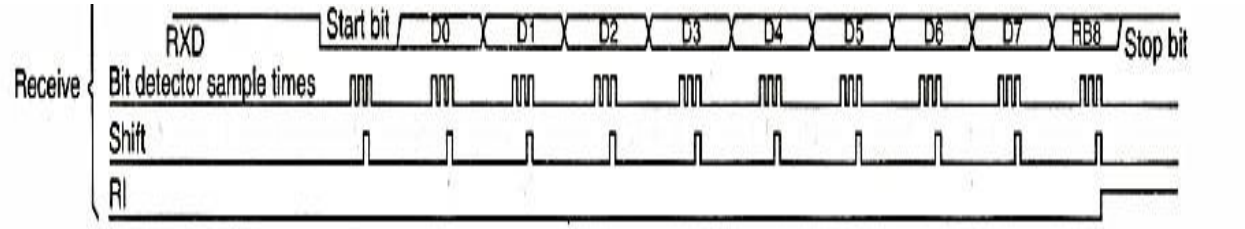



Figure 7: mode 2 and mode 3

**Serial Control Register (SCON)**

- This special function register (SFR) controls the operations of the serial port.
- This register is used to define the operatingmodes.
- This also receives the 9th bit and contains the transmit and receive interrupt flags aswell.

Serial Control Register (SCON):

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI |

I /O PORTS

Each port of 8051 has bi-directional capability.

Port 0 is called 'true bidirectional port' as it floats (tri-stated) when configured as input. Port-1, 2, 3 are called 'quasi bidirectional port'**.**

- To communicate data with the external world the microcontroller needs ports.
- The ports may support either parallel or serial data transfer.
- It has 4 I/O ports namely, **Port 0, Port 1, Port 2 & Port 3**
- Port 1: is exclusively for input & output functions.
- Port 0, 2 & 3: perform functions other than parallel data transfer.

- All 4 ports are bidirectional.
- The 8 port pins are connected through 8 D type port latches.

PORT 0:

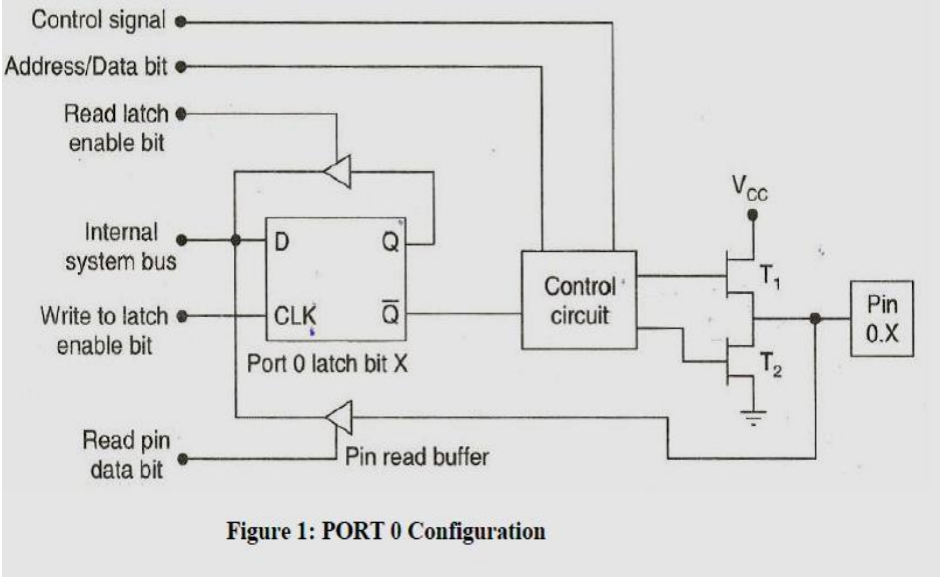



**Figure 1: PORT 0 Configuration**

Port -0 has 8 pins (P0.0-P0.7).

- Port -0 has 8 pins (P0.0-P0.7).
- Port-0 can be configured as a normal bidirectional I/O port or it can be used for Address / data interfacing for accessing external memory.
- When control is '1', the port is used for address/data interfacing. When the control is '0', the port can be used as a normal bi-directional I/O port.
- Port-0 latch is written to with 1's when used for external memory access.

**Port 1:**

- Port-1 has 8 pins (P1.1-P1.7)
- Port-1 does not have any alternate function i.e. it is dedicated solely for I/O interfacing.
- When used as output port, the pin is pulled up or down through internal pull-up.
- To use port-1 as input port, '1' has to be written to the latch. In this input mode when '1' is written to the pin by the external device then it read fine.

• But when '0' is written to the pin by the external device then the external source must sink current due to internal pull-up.

• If the external device is not able to sink the current the pin voltage may rise, leading to a possible wrong reading.

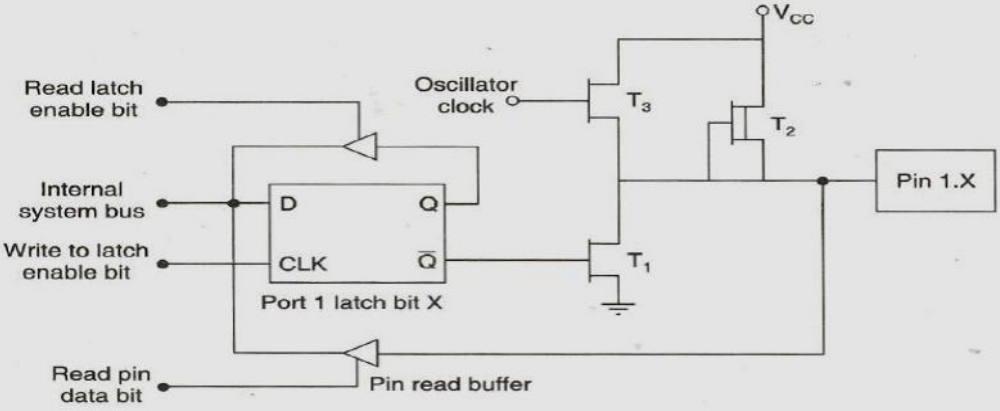



**Figure 2: PORT 1 Configuration**

**Port 2 :**

• Port-2 has 8-pins (P2.0-P2.7)

• Port-2 is used for higher external address byte or a normal input/output port.

• The I/O operation is similar to Port-1. Port-2 latch remains stable when Port-2 pin are used for external memory access.

• Here again due to internal pull-up there is limited current driving capability.

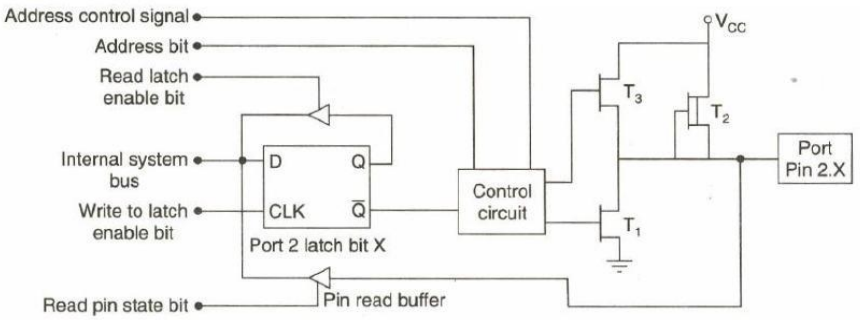



**Figure 3: PORT 2 Configuration**

2

## Port 3:

- Port-3 has 8 pin (P3.0-P3.7)
- Port-3 pins have alternate functions.
- Each pin of Port-3 can be individually programmed for I/O operation or for alternate function.
- The alternate function can be activated only if the corresponding latch has been written to '1'.
- To use the port as input port, '1' should be written

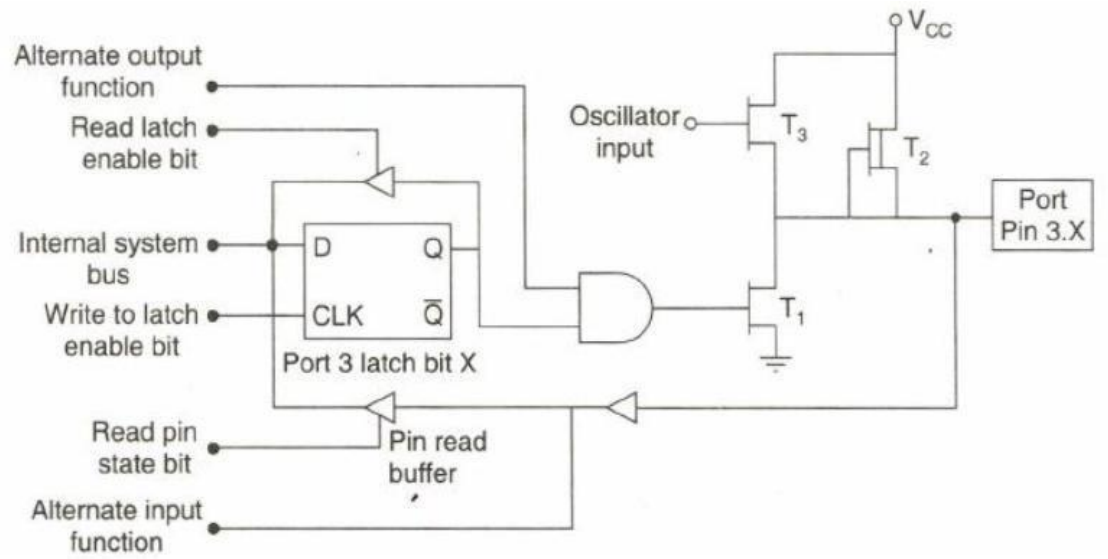



**Figure 3: PORT 2 Configuration**

**Alternate Functions of Port 3:**

- P3.0 and P3.1 are used for the RxD (Receive Data) and TxD (Transmit Data) serial communications signals.
- Bits P3.2 and P3.3 are meant for external interrupts.
- Bits P3.4 and P3.5 are used for Timers 0 and 1 and P3.6 and P3.7 are used to provide the write

and read signals of external memories connected in 8031 based systems

| S.No | Port 3 bit | Pin No | Function |
|---|---|---|---|
| 1. | P3.0 | 10 | RxD |
| 2. | P3.1 | 11 | TxD |
| 3. | P3.2 | 12 | $\overline{\text{INT0}}$ |
| 4. | P3.3 | 13 | $\overline{\text{INT1}}$ |
| 5. | P3.4 | 14 | T0 |
| 6. | P3.5 | 15 | T1 |
| 7. | P3.6 | 16 | $\overline{\text{WR}}$ |
| 8. | P3.7 | 17 | $\overline{\text{RD}}$ |

**Table1: PORT 3 alternate functions**