TIMER SECTION OF 8051

8051 has **2, 16-bit Up Counters** T1 and T0.
If the counter **counts internal clock** pulses it is known as **timer**.
If it counts **external clock** pulses it is known as **counter**.
Each counter is divided into 2, 8-bit registers TH1 - TL1 and TH0 - TL0.
The **timer** action is **controlled** mainly by the **TCON** and the **TMOD** registers.

TCON - Timer Control (SFR) [Bit-Addressable As TCON.7 to TCON.0]

TF1	**TR1**	**TF0**	**TR0**	IE1	IT1	IE0	IT0

TF1 and TF0: (Timer Overflow Flag)
Set (1) when Timer 1 or Timer 0 overflows respectively i.e. its bits roll over from all 1's to all 0's.
Cleared (0) when the processor executes ISR (address 001BH for Timer 1 and 000BH for Timer 0).

TR1 and TR0: (Timer Run Control Bit)
Set (1) - Starts counting on Timer 1 or Timer 0 respectively.
Cleared (0) - Halts Timer 1 or Timer 0 respectively.

IE1 and IE0: (External Interrupt Edge Flag)
Set (1) when external interrupt signal received at $\overline{\textbf{INT1}}$ or $\overline{\textbf{INT0}}$ respectively.
Cleared (0) when ISR executed (address 0013H for Timer 1 and 0003H for Timer 0).

IT1 and IT0: (External Interrupt Type Cobtrol Bit)
Set (1) - Interrupt at $\overline{\textbf{INT1}}$ or $\overline{\textbf{INT0}}$ must be -ve edge triggered.
Cleared (0) - Interrupt at $\overline{\textbf{INT1}}$ or $\overline{\textbf{INT0}}$ must be low-level triggered.

TMOD - Timer Mode Control (SFR) [NOT Bit-Addressable]

Gate	C/T	M1	M0	Gate	C/T	M1	M0
Timer 1				Timer 0			

C/T: (Counter/Timer)
Set (1) - Acts as Counter (Counts external frequency on T1 and T0 pin inputs).
Cleared (0) - Acts as Timer (Counts internal clock frequency, fosc/12).

Gate: (Gate Enable Control bit)
Set (1) - Timer controlled by hardware i.e. INTX signal.
Cleared (0) – Counting independent of INTX signal.

M1, M0: (Mode Selection bits)
Used to select the operational modes of the respective Timer.

M1 M0	Timer Mode
0 0	Mode 0
0 1	Mode 1
1 0	Mode 2
1 1	Mode 3

Timer Counter Interrupts

To use the timer, a certain **count value** is placed **in** the **Count Register**.

This value is the **Max Count - Desired Count + 1**.

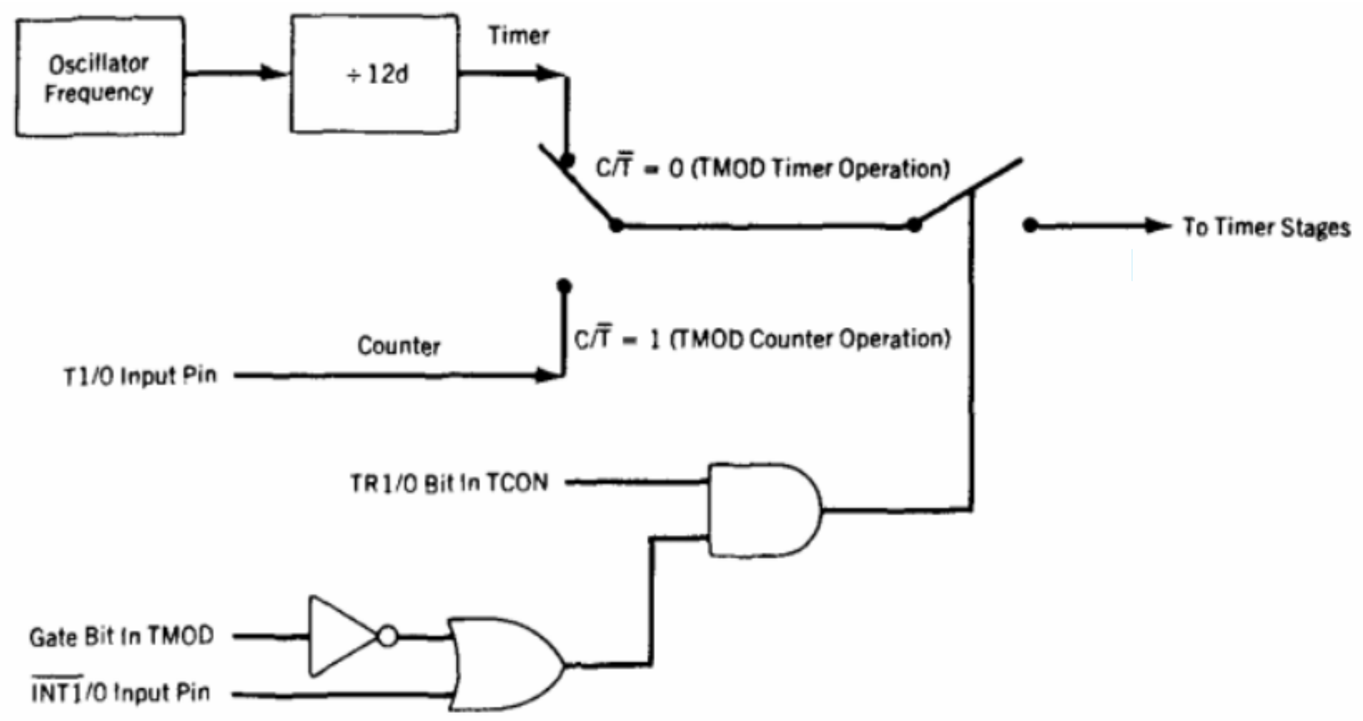
On **each count** (rising edge of the input clock) the **counter increments** its value.

When the counter rolls over (i.e. form **all 1's to all 0's**) it is said to **overflow**.

Thus the Timer Overflow Flag, **TFX** (TF1 or TF0) is **set**.

If timer **interrupt** is **enabled** then the **Timer Interrupt will occur** on overflow.

Timer Counter Logic


Oscillator Frequency
÷ 12d
Timer
C/$\overline{T}$ = 0 (TMOD Timer Operation)
To Timer Stages
Counter
T1/0 Input Pin
C/$\overline{T}$ = 1 (TMOD Counter Operation)
TR1/0 Bit In TCON
Gate Bit In TMOD
$\overline{INT1}$/0 Input Pin

As shown above, based on **C/T** bit the timer functions as a **Counter** or as a **Timer**.

If it is a **Timer,** it will count the **internal clock frequency** of 8051 **divided by 12_d** (f/12).

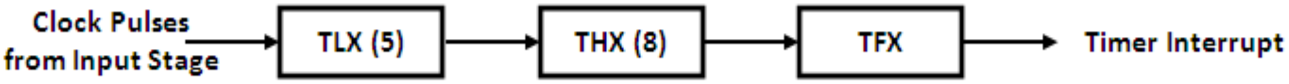
If it is a **Counter,** the **input clock signal** is applied at the **TX** (T1 or T0) input pins for Timer1 or Timer0 respectively. #Please refer Bharat Sir's Lecture Notes for this ...

As shown the **Timer** is **running only if** the **TRX** bit (TR1 or TR0) **is set**.

Also **if** the **Gate** bit is **set** in the TMOD then the **INTX** ($\overline{\textbf{INT1}}$ or $\overline{\textbf{INT0}}$) pin **must be "high (1)"** for the timer to count.

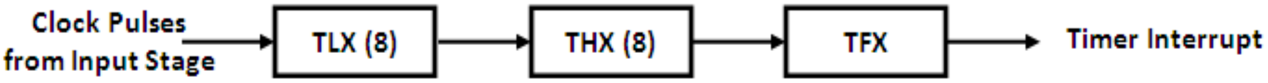
Timer Modes

a) **Timer Mode 0** (13-bit Timer/Counter)



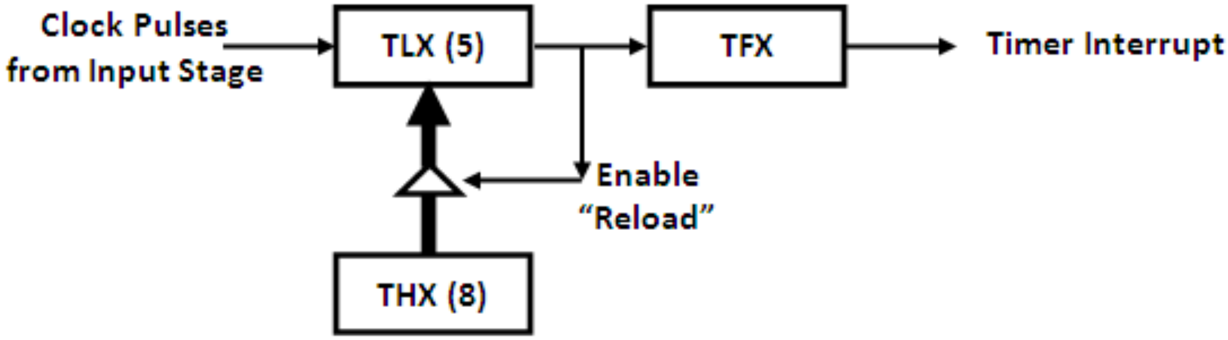
THX is used as an **8-bit counter**.
TLX is used as a **5-bit pre-set**. Hence **13-bits** are used for counting.
On **each count** the **TLX increments**.
Each time **TLX rolls-over**, **THX increments**.
Thus the **input frequency** is **divided by 32** (5-bits of TLX and **2^5 = 32**).
The timer overflow flag **TFX** is **set** only **when THX overflows** i.e. rolls from FFH to 00H.
Max Count = 2^{13} = 8K = 8192 (1FFFH). Hence **Max Delay ➔ 8192(12/f)**

b) **Timer Mode 1** (16-bit Timer/Counter)



All **16-bits** of the **Counter** are used (8 bits of THX and 8 bits of TLX).
On **each count** the 16-bit **Timer increments**.
The timer overflow flag **TFX** is **set when** the **Timer rolls-over from FFFFH to 0000H**.
Max Count => 2^{16} = 16K = 65536 (FFFFH). Hence **Max Delay ➔ 65536(12/f)**.

c) **Timer Mode 2** (Auto reload TL from TH)

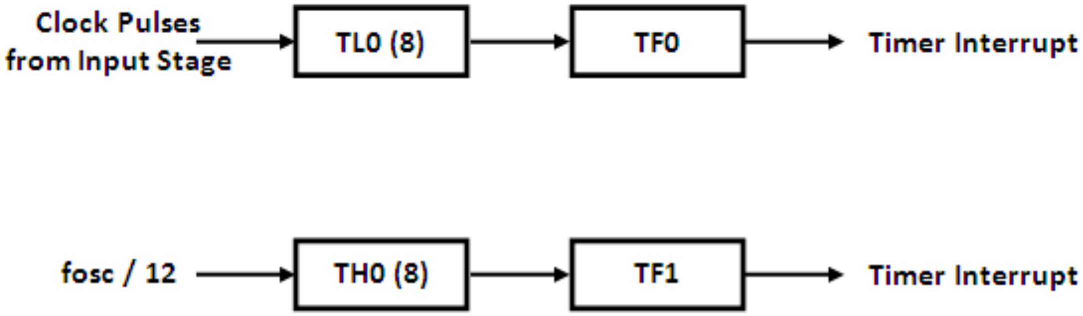


TLX is used as an **8-bit counter**.
THX holds the **count value** to be **reloaded**.
On **each count TLX increments**.
When **TLX rolls-over** (i.e. from FFH to 00H), the following events take place:

1. Timer overflow flag **TFX** is **set**, hence timer interrupt occurs.
2. The value of **THX** is **copied into TLX**. Hence TLX is **auto-reloaded** form THX, and the process repeats.

Thus the timer interrupt occurs at regular intervals "**Continuously**".
This mode is used to generate a desired frequency using the Timer Flag.
Max Count ➔ 2^8 = 256 (FFH). Hence **Max Delay ➔ 256(12/f)**.

d) **Timer Mode 3** (Two 8-bit Timers Using Timer0)



Timer 0 is used **as 2 separate 8-bit timers TH0 and TL0**.
TL0 uses the control bits (**TR0 and TF0**) of Timer 0.
It can work as a **Timer or** a **Counter**.
TH0 uses the control bits (**TR1 and TF1**) of Timer 1.
It can work only as a **Timer.** #Please refer Bharat Sir's Lecture Notes for this ...
Timer 1 can be in Mode 0, Mode 1, or Mode 2, but will not generate an interrupt.