TIMER AND COUNTER PROGRAMMING

The 8051 has two times / counters. They can be used either as timers to generate a time delay or as counters to count events happening outside the micro-controller. The two timers in 8051 are:

- Timer 0 and
- Timer 1.

Basic registers of the timer

Both Timer 0 and Timer 1 are 16 bit wide. Since 8051 has an 8-bit architecture, each 16-bit timer is accessed as two separate registers of low byte and high byte.

Timer 0 registers

The 16-bit register of Timer 0 is accessed as low byte and high byte. The low byte register is called TL0 (Timer 0 low byte) and the high byte register is referred to as TH0 (Timer 0 high byte).

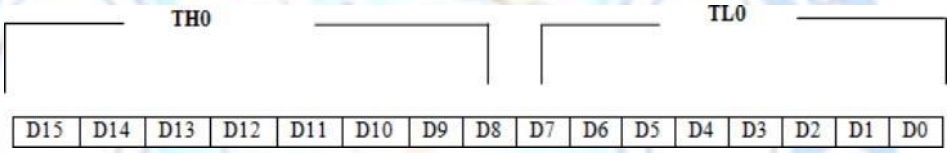

Fig. Timer 0 registers

Timer 1 registers

The 16- bit register of timer 1 is accessed as low byte and high byte. The low byte register is called TL1 (Timer 1 low) and the high byte register is called TH1 (Timer 1 high)

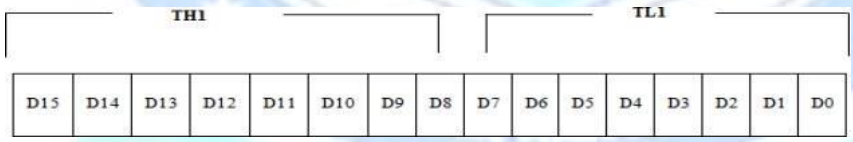

Fig. Timer 1 registers

TMOD (timer mode) register

TMOD is an 8 bit register in which the lower 4 bits are set for Timer 0 and the higher 4 bits are set for Timer 1.

(MSB) (LSB)

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

Timer1 Timer 0

Fig: TMOD Register

GATE

When GATE = 0, the start and stop of timer are controlled by software by means of TR (timer start) bits TR0 and TR1. The timer is started by using the instruction "SETB TR0" for timer 0 and "SETB TR1" for timer 1 and stop by using the instruction "CLR TR1" for Timer 1, and "CLR TR0" for timer 0.

When GATE = 1, the start and stop of timer are controlled by hardware.

C/$\overline{T}$

This bit in the TMOD register is used to decide whether the timer is used as a delay generator or an event counter. If C/$\overline{T}$ = 0, it is used as a timer. If C/$\overline{T}$ = 1, it is used as a counter.

M1, M0: (mode select bits)

M1	M0	Mode	Operating Mode
0	0	1	13 bit timer mode
0	1	1	16 bit timer mode
1	0	2	8 bit timer with auto reload
1	1	3	Split timer mode

Timer operating modes

Mode 0 (13 bit timer)

In this mode, TL0 and TH0 for timer 0 are used as 13 bit register, i.e., all the 8 bits of TL0 and 5 bits in TH0. The 13-bit counter can hold values between 0000 to 1FFFH in TH - TL. Therefore, when the timer reaches the maximum of 1FFH, it rolls over to 0000, and TF is raised.

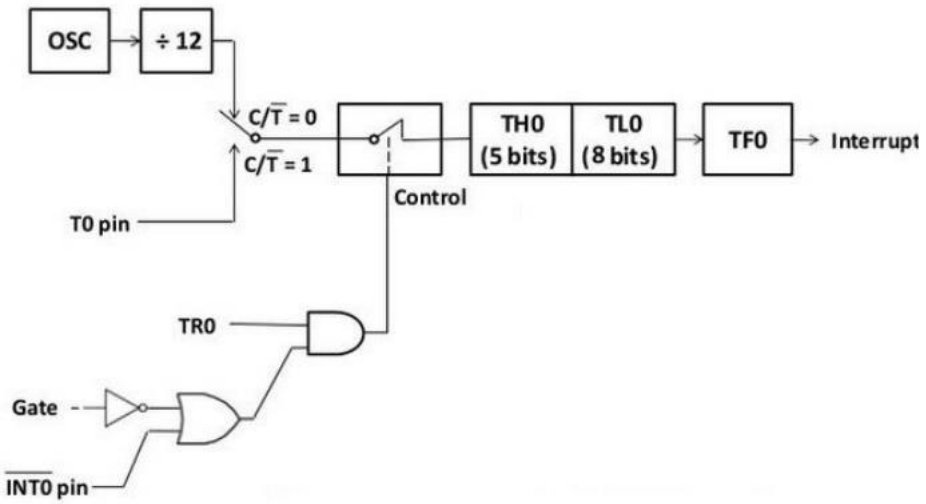

Fig: Mode 0

If C/$\overline{T}$=0, then the register is incremented after every machine cycle, i.e, at the rate of 1/12th of the oscillator frequency. For the register to get incremented the control switch should be closed. This switch is logic controlled.

There are many ways in which the logic can be implemented.

- Case 1

TR0 = 1 (high)

Gate = 0 (low) and $\overline{INT0}$ = 0 (low)

- Case 2

TR0 = 1 (high)

Gate = 0 (high) and $\overline{INT0}$ = 1 (high)

Mode 0 programming

The following are the characteristics and operations of mode 0:

1. It is a 13-bit timer; therefore, it allows values of 0000 to 1FFFH to be loaded into the timer‟s registers TL and TH.
2. After TH and TL are loaded with a 13-bit initial value, the timer must be started. This is done by “SETB TR0” for Timer 0 and SETB TR1” for Timer 1.
3. After the timer is started, it starts to count up. It counts up until it reaches its limit of $1FFF_H$. When it rolls over from $1FFF_H$ to 0000, it sets the timer flag to 1. When the timer flag is raised, one option to stop the timer is by using the instructions “CLR TR0” for timer 0 and “CLR TR1”, Timer 1.
4. After the timer reaches its limit and rolls over, in order to repeat the process, the registers TH and TL must be reloaded with the original value, and TF must be reset

Mode 1 (16 bit timer)

In mode 1, 16 bit timer is used. The 16-bit counter can hold values between 0000 to FFFFH in TH - TL. Therefore, when the timer reaches its maximum of FFFH, it rolls over to 0000, and TF is raised.

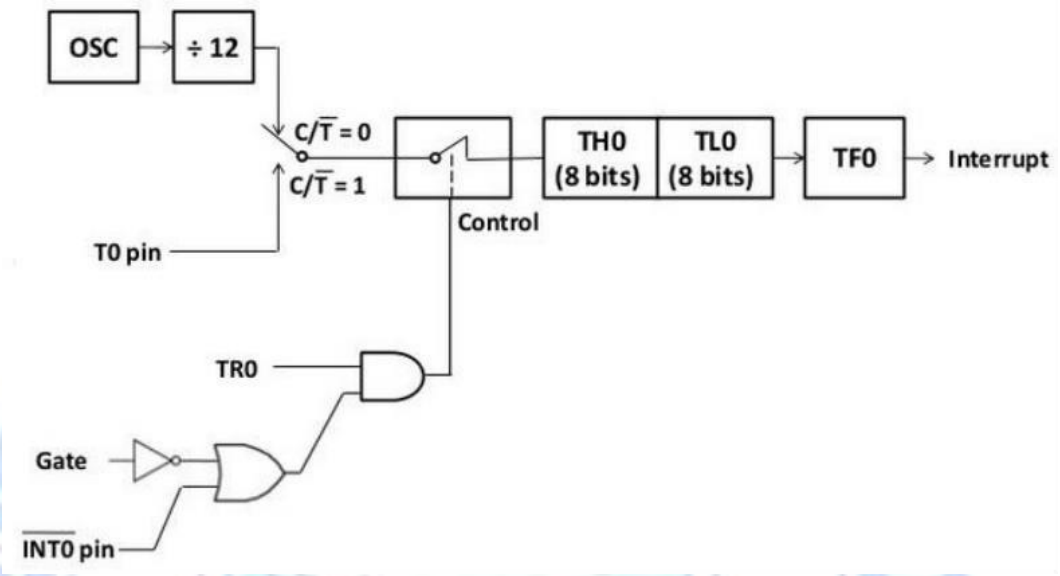

Fig: Mode 1

Mode 1 programming

The characteristics and operations of mode 1 are:

1. It is a 16-bit timer; therefore, it allows values of 0000 to $FFFF_H$ to be loaded into the timer‟s registers TL and TH.
2. After TH and TL are loaded with a 16-bit initial value, the timer must be started. This is done by “SETB TR0” for Timer 0 and SETB TR1” for Timer 1.
3. After the timer is started, it starts to count up. It counts up until it reaches its limit of $FFFF_H$. When it rolls over from $FFFF_H$ to 0000, it sets timer flag to 1. When the timer flag is raised, one option to stop the timer is using the instructions “CLR TR0” for timer 0 and “CLR TR1”, for Timer 1.
4. After the timer reaches its limit and rolls over, in order to repeat the process the registers. TH and TL must be reloaded with the original value, and TF must be reset to 0.

Mode 2 (8 – bit with auto reload)

In this mode, the timer register is 8 bits wide. TL0 for Timer 0 or TL1 for timer 1 is used for this purpose. This mode is also called the auto-reload mode as the timer generates an interrupt on overflow and after generating the interrupt, it reloads the preset value from TH0 into TL0. This preset value is put into TH0 through software.

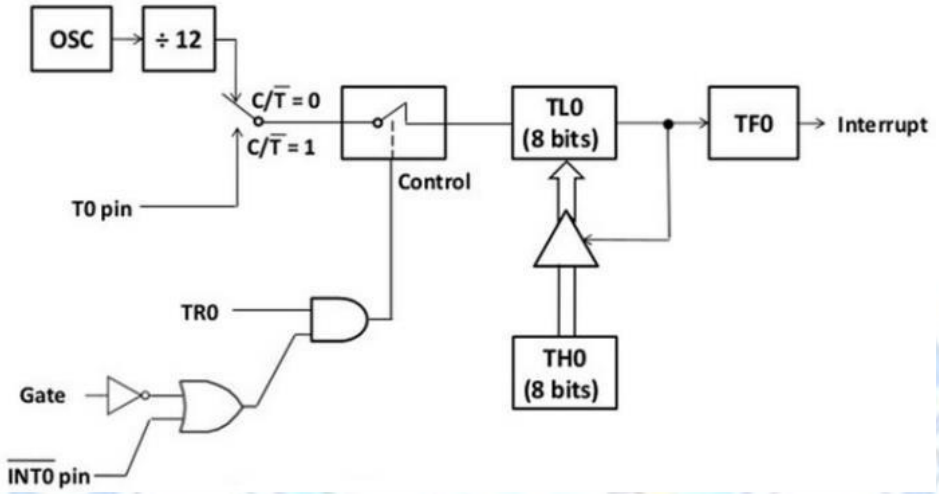

Fig: Mode 2- 8 bit timer with auto reloaded

Mode 2 programming

The following are the characteristics and operations of mode 2.

1. It is an 8-bit timer; therefore it allows only values from 00 to FFH to be loaded into the timer‟s register TH.

2. After TH is loaded with the 8-bit value, the 8051 gives a copy to TL. And the timer is started. This is done by the instruction “SETB TR0” for Timer 0 and “SETB TR1, for Timer 1.

3. After the timer is started, it starts to count up by incrementing the TL registers. It counts up until it reaches its limit of FFH. When it rolls over from FFH to 00, it sets high the TF (timer flag). If Timer 0 is used TFO goes high; if Timer 1 is used TF1 is raised.

4. When the TL registers rolls from FFH to 00 and TF is set to 1, TL is reloaded automatically with the original value kept by the TH register.

To repeat the process, clear TF and without any programmer the original value is reload.

Mode 3 (Split timer mode)

If the Timer 0 is put into mode 3, then it acts as two 8-bit counters (TL0 and TH0 become two separate counters). All the timer 0 control bits (C/$\overline{T}$, Gate, TR0, TF0 and $\overline{INT0}$

) are used by TL0 itself and TH0 register is locked into a timer function. TH0 is counting machine cycles and has taken over the use of TR1 and TF1 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 TR1 = 0, hence opening the switch.

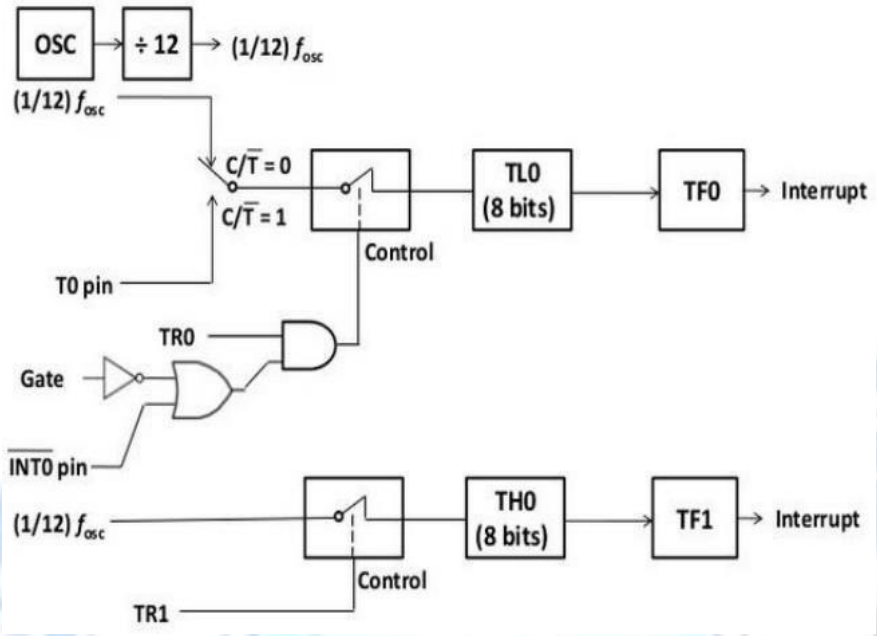

Fig : Mode 3

The start and stop of the timer is controlled by TR bit in timer control (TCON) Register.

TCON register

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Fig: TCON Register

TF1: Timer 1 overflows 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.

IE1: 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 $\overline{INT1}$ is falling edge triggered
- When IT0 = 0 then $\overline{INT1}$ 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 specify the falling edge/low level triggered external interrupts.

- When IT0 = 1 then $\overline{INT0}$ is falling edge triggered
- When IT0 = 0 then $\overline{INT0}$ is low-level triggered