The 8051 Timer/Counter and Serial port

8051 Timer/Counter and Serial Port Operation

- Introduction
- TCON
- TMOD
- Mode 0, 1, 2 & 3 operations
- Serial port operation
- Mode operation for serial port
- Power saving modes

Timers

Many microcontroller applications require the counting of external events, such as :

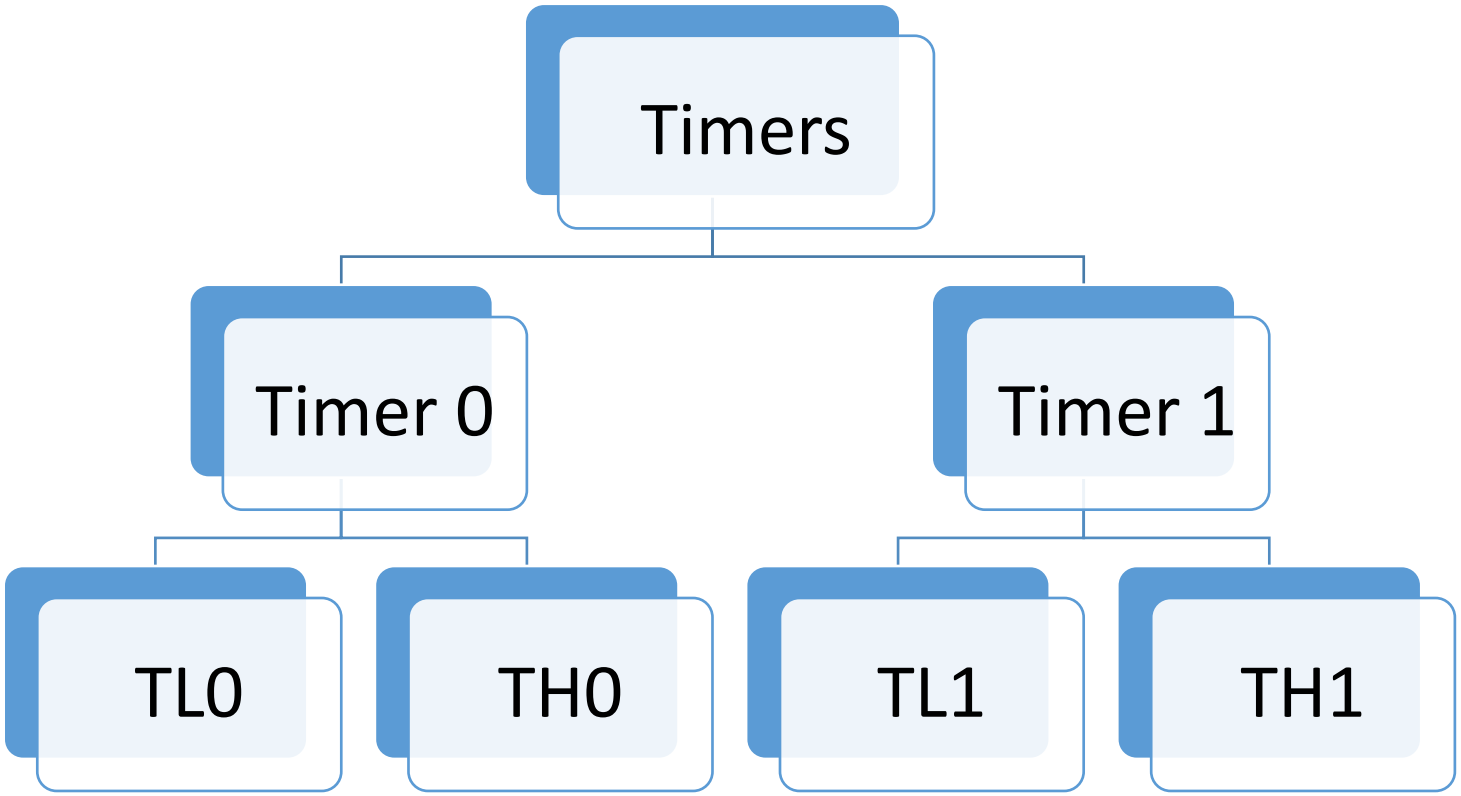
- the frequency of a pulse train
- the generation of precise internal time delays between computer actions.

Basic function of timers

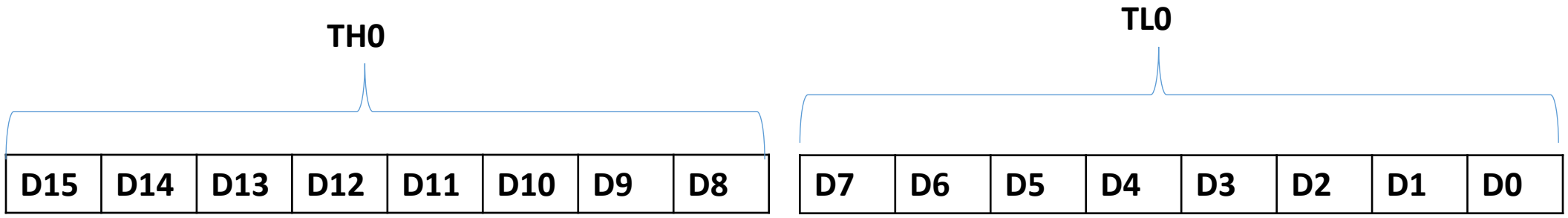
1. To measure the time-calculating time elapsed between events. This capability of timer is deployed in generating time delays as well as waveforms.
2. To count events or to measure frequency or pulse width of an unknown signal.
3. To provide the clock signal to other circuits

Timer Types

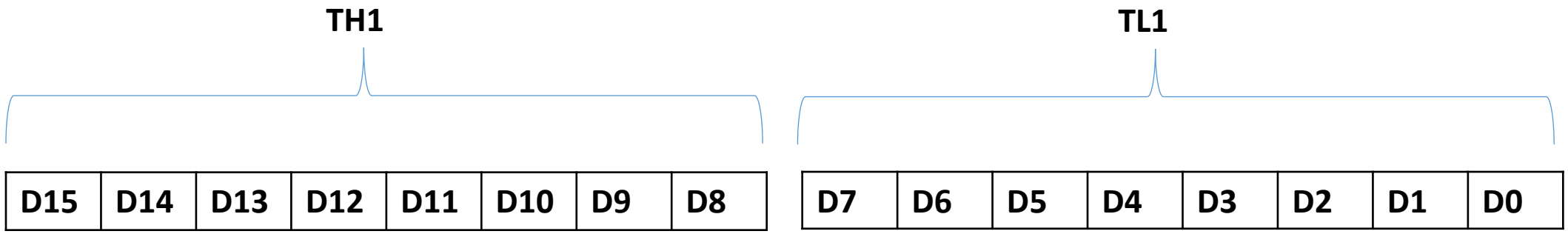
The 8051 has two 16-Bit timers :-Timer 0 & Timer 1.



Timer 0 Registers



Timer 1 Registers



Timers

Both timers consist of two 8 bit registers (THx and TLx).

- Both share the timer control (TCON) register and timer mode (TMOD) register.
- A timer/counter can be used to create time delay, or as a counter to count external events happening outside the microcontroller that occurred within a time interval.
- The only difference between counter and timer is the source of clock pulse.
- When used as a timer, the clock pulse is obtained from oscillator through divide by 12 d circuit.
- When used as a counter pin T0 (P3.4) supplies pulses to counter 0 and pin T1 (P3.5) supplies pulses to counter 1.

The timers are programmed with 6 SFR

- TCON Timer Control Register (Timer Enable bits and Timer Flags)
- TMOD Timer Mode Register (set Timer mode, selects int/ext pulses)
- TL0, TL1 Low 8 bits of Timers 0 and 1
- TH0, TH1 High 8 bits of Timers 0 and 1

An interrupt is the occurrence of a condition--an event -- that cause a temporary suspension of a program while the event is serviced by another program (Interrupt Service Routine ISR or Interrupt Handler).

Timer Control Register (TCON)

Bit 7 Bit 0

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	**Timer 1 overflow flag set when timer/counter overflows** over from FFFFH to 0000H Cleared when processor begin to execute ISR(ISR address=001B H)
TR1	**Timer 1 run control bit** Set = Timer ON and Clear = Timer OFF
TF0	**Timer 0 overflow flag set when timer/counter overflows** over from FFFFH to 0000H Cleared when processor begin to execute ISR(ISR address=000B H)
TR0	**Timer 0 run control bit** Set = Timer ON and Clear = Timer OFF
IE1	**External Interrupt 1 edge flag set when processor receive interrupt signal at $\overline{\text{INT}}$ 1**
IT1	**Timer 1 interrupt control bit; 0 = low level triggered, 1= falling edge triggered**
IE0	**External Interrupt 0 edge flag set when processor receive interrupt signal at $\overline{\text{INT}}$ 0**
IT0	**Timer 0 interrupt control bit; 0 = low level triggered, 1= falling edge triggered**

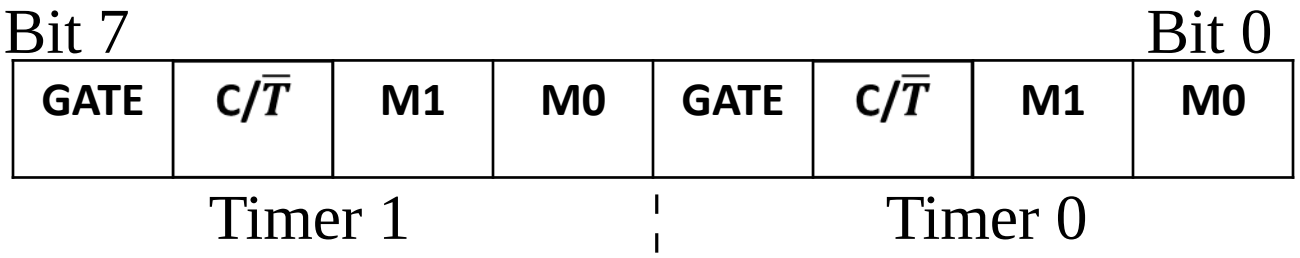
Timer Control Register (TCON)

- Timer control register controls the timer/counter operation.
- This register is an 8-bit register.
- TCON is bit addressable.
- The address of TCON is 88H.
- It is partly related to Timer and partly to interrupt.

Timer Mode Control (TMOD) Register

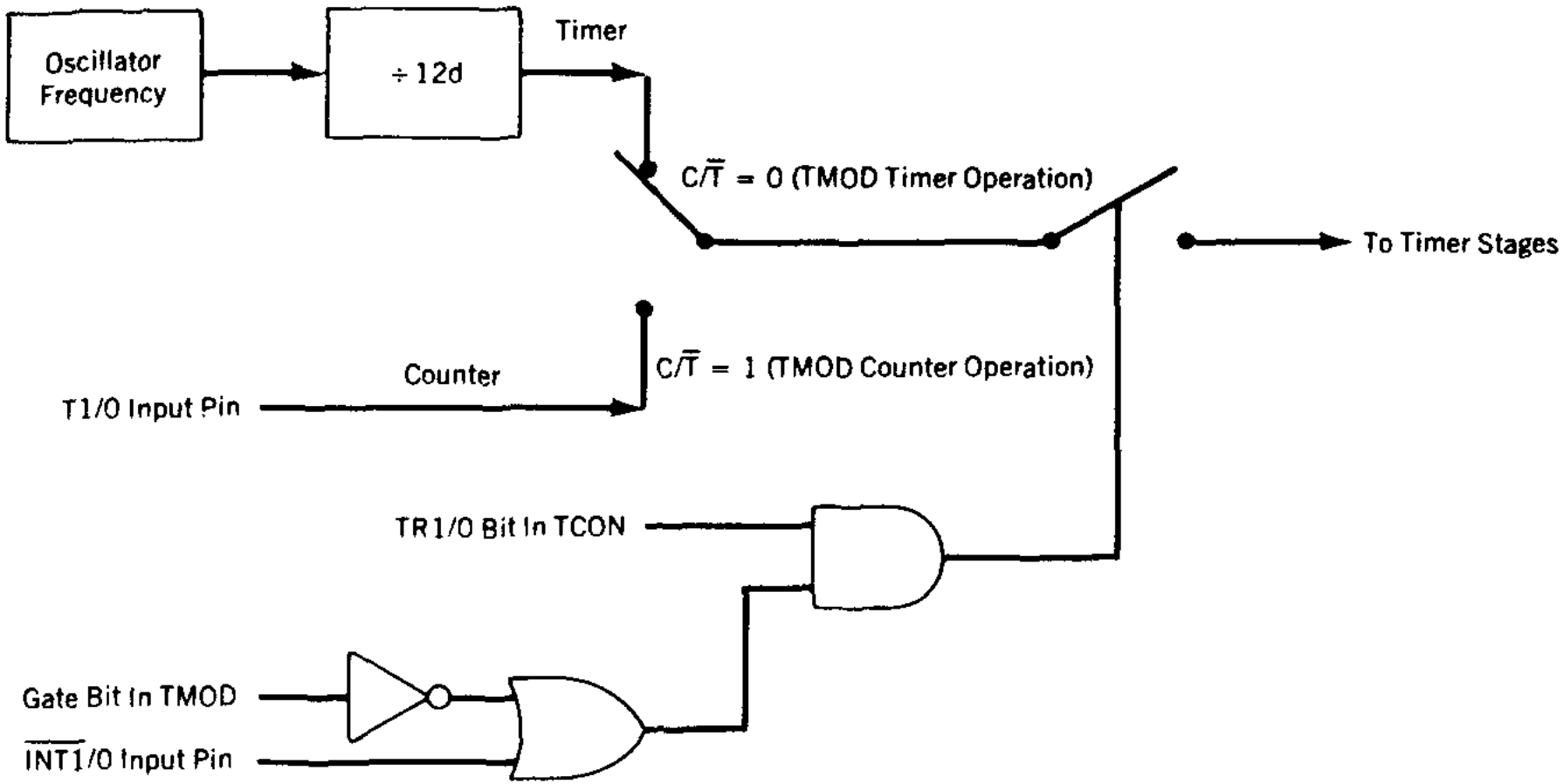
- Lower 4-bits are used for controlling Timer 0 and Upper 4-bits are used for controlling Timer 1.
- $C/\overline{T}$ selects timer or counter operation and M1 & M0 select the mode.

Timer Mode Control (TMOD) Register



GATE	If GATE=0, Timer 0 Or 1 is enabled if TR0 Or TR1 control bit is set. If GATE=1, Timer 0 Or 1 is enabled if $\overline{INT}$**0**=1 and TR0=1 OR $\overline{INT}$**1=1** and TR1=1		
C/$\overline{\boldsymbol{T}}$	0 = Timer mode ; 1 = Counter mode		
M1 M0 Timer mode selection	**M1**	**M0**	**Mode**
	0	0	Mode 0
	0	1	Mode 1
	1	0	Mode 2
	1	1	Mode 3

Timer/Counter Control Logic



Timer/Counter Control Logic

- If a counter is programmed to be timer, it will count the internal clock frequency of 8051 oscillator divided by 12d.
- The resultant timer clock is gated to the timer .
- IN order for oscillator clock pulses to reach the timer, C/$\overline{T}$ bit in TMOD register set to 0(Timer Operation).
- Bit TRX in TCON register set to 1 (timer run) & gate bit in TMOD register must ne 0,or external pin $\overline{INT}$X=1 .
- Counter is configured as a timer ,then timer pulses are gated to the counter by run bit & gate bit or external i/p bits $\overline{INT}$X.

Timer Mode 0

- In this mode, the timer is used as a 13-bit UP counter as follows.

Timer Mode 0 13-Bit Timer/Counter

Timer Mode 0

- The lower 5 bits of TLX and 8 bits of THX are used for the 13 bit count.
- Upper 3 bits of TLX are ignored.
- When the counter rolls over from all 0's to all 1's, TFX flag is set and an interrupt is generated.
- The input pulse is obtained from the previous stage.
- If TR1/0 bit is 1 and Gate bit is 0, the counter continues counting up.
- If TR1/0 bit is 1 and Gate bit is 1, then the operation of the counter is controlled by $\overline{INT}X$ input.
- This mode is useful to measure the width of a given pulse fed to $\overline{INT}$X input.

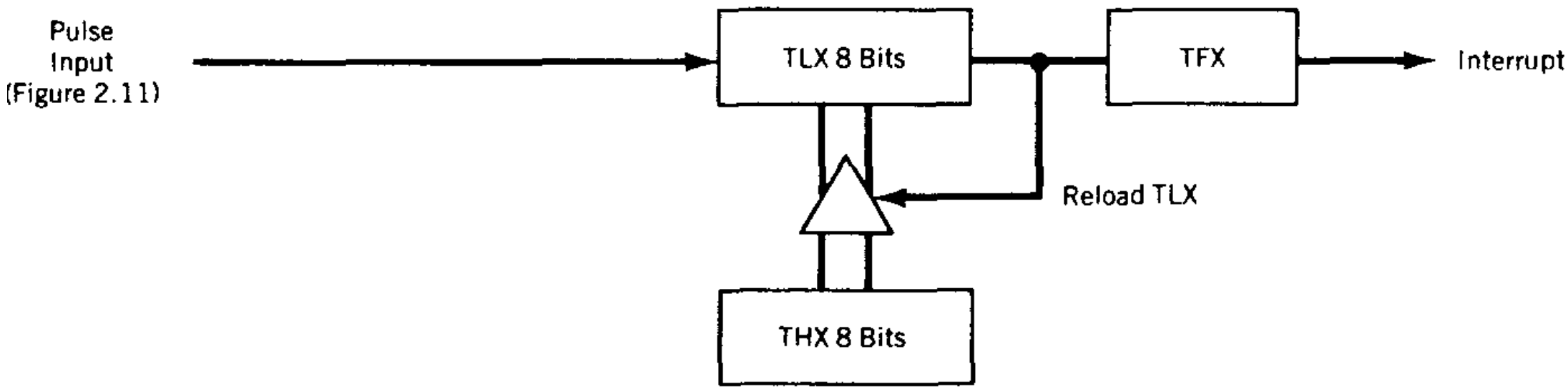
Timer Mode 1

- Simple 16-bit counter.
- The Gate input controls whether the Counter runs while gated by the interrupt signal or not.

Timer Mode 1 16-Bit Timer/Counter

Timer Mode-2: (Auto-Reload Mode)

This is a 8 bit counter/timer operation.



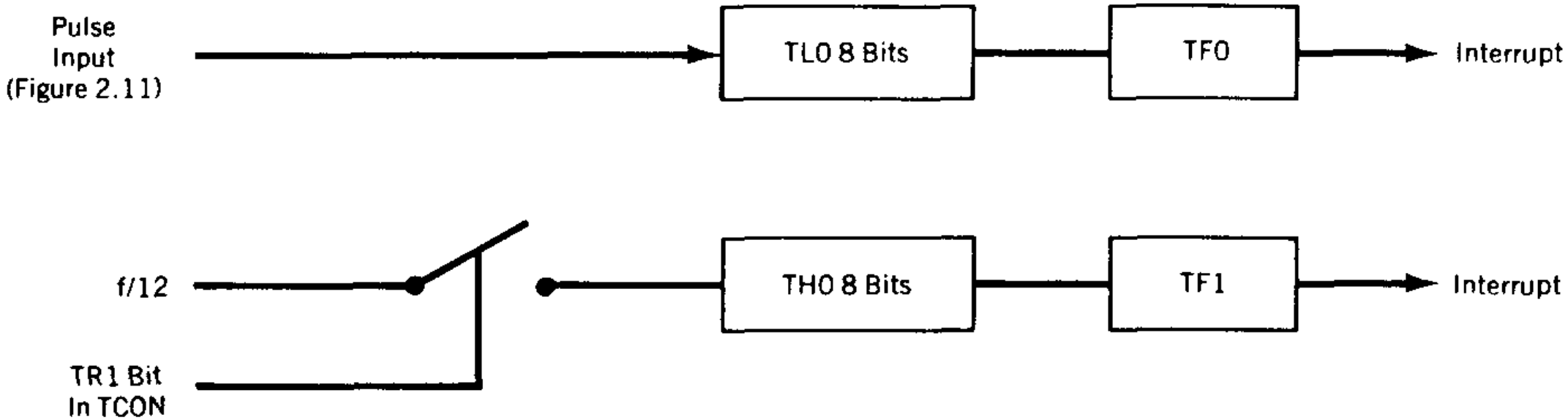
Timer Mode 2 Auto - Reload of TL from TH

Timer Mode-2: (Auto-Reload Mode)

- Counting is performed in TLX while THX stores a constant value.
- In this mode when the timer overflows i.e. TLX becomes FFH, it is fed with the value stored in THX.
- For example if we load THX with 50H then the timer in mode 2 will count from 50H to FFH.
- After that 50H is again reloaded.
- This mode is useful in applications like fixed time sampling.

Timer Mode 3

- Timer 1 in mode-3 simply holds its count.
- The effect is same as setting TR1=0. Timer0 in mode-3 establishes TL0 and TH0 as two separate counters.

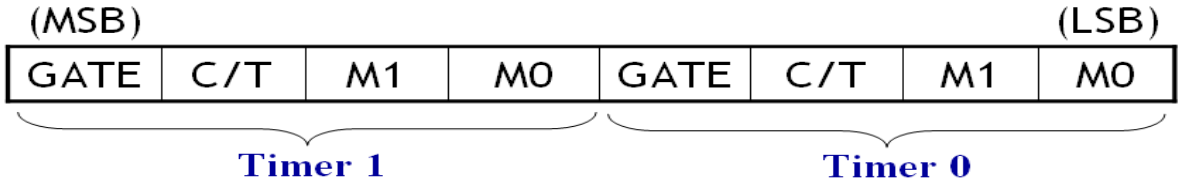


Timer Mode 3 Two 8 - Bit Timers Using Timer 0

Timer Mode 3

- Control bits TR1 and TF1 are used by Timer-0 (higher 8 bits) (TH0) in Mode-3 while TR0 and TF0 are available to Timer-0 lower 8 bits(TL0).

Timers Programming



Example 8-1

Indicate which mode and which timer are selected for each of the following.

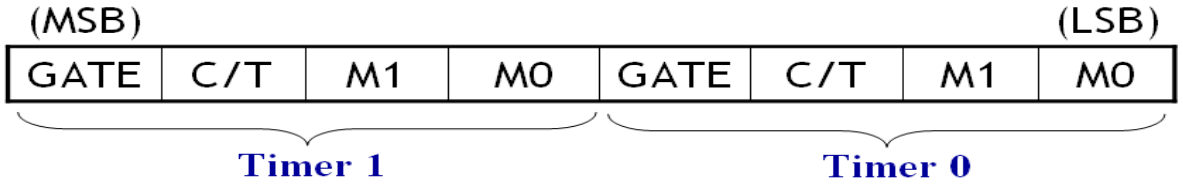
(a) MOV TMOD, #01H
(b) MOV TMOD, #20H
(c) MOV TMOD, #12H

Solution:

Convert the values from hex to binary:

(a) TMOD = 0000 0001, mode 1 of Timer 0 is selected
(b) TMOD = 0010 0000, mode 2 of Timer 1 is selected
(c) TMOD = 0001 0010, mode 2 of Timer 0, and mode 1 of Timer 1are selected

Timers Programming



Example 8-2

1. Write instructions to do the followings:
    a. Set Timer 0 in mode 1, use 8051 XTAL for the clock source, instructions to start and stop the timer,
    b. Set value F0ABH to Timer 0.
    c. Start Timer 0
2. Determine the time for Timer 0 rolling over if XTAL = 12 MHz.

Solution:
(1)

```
MOV   TMOD, #01H          ;  Timer 0  and mode 1 set,  C/T = 0 to use XTAL                    ;
    clock source, Gate = 0 to use software ON/OFF
MOV   TL0, #0ABH          ; TL0 =ABH
MOV   TH0, #0F0H          ; TH0 =F0H
SETB  TR0                 ; Start Timer 0
```

Timers Programming

THx TLx	# of count to set TFx
FFH FFH	1
FFH FEH	2

Solution:

(2)

Assume 12 MHz clock :

Time for 1 Timer clock = 1 machine cycle = $\frac{1}{12\text{x}10^6}\text{x}12\text{ s} = 1\mu\text{s}$

Counts for Timer rolling over = Counts from F0ABH to FFFFH
plus rolling over to 0

Timer clock cycles = (FFFFH – F0ABH + 1) = 0F55H

65536_{10} – F0ABH = 3925 in decimal

∴ Time for Timer 0 rolls over = 3925 x 1 μs = 3925 μs #

Timers Programming

Ie. Toggle P1.3 every 10ms

Example 8-3

Assume XTAL = 12 MHz, write a program to generate a square wave of 50 Hz frequency on pin P1.3 by using timer 1 as time control.

Solution:

The period of the square wave, T = 1/50 Hz = 20 ms

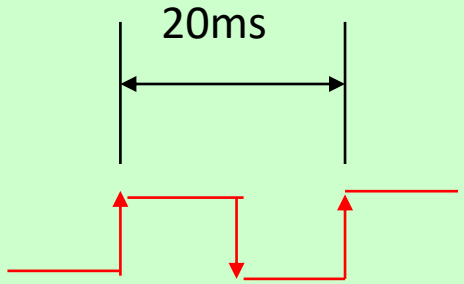
½ of it for the high and low portions of the pulse = 10 ms

10 ms/ 1μs = 10,000 timer cycles are needed for each pulse.

Timer 1 value to be set = 65536 – 10000 = 55536 in decimal = D8F0H

i.e. TH1 = D8H and TL1 = F0H **(need 16 bit Counter, so take mode 1)**

65535 – 10000 + 1




```
        MOV   TMOD, #10H     ;  Timer 1, mode 1
AGAIN:  MOV   TL1, #0F0H     ;  TL1 =F0H
        MOV   TH1, #0D8H     ;  TH1 =D8H
        SETB  TR1   ;  Start Timer 1
BACK:   JNB   TF1, BACK      ;  Stay until timer rolls over
        CLR   TR1            ;  Stop Timer 1
        CPL   P1.3                    ;  Complement P1.3 to set Hi, Low
        CLR   TF1            ;  Clear Timer flag by software
        SJMP  AGAIN
```