TIMER PROGRAMMING

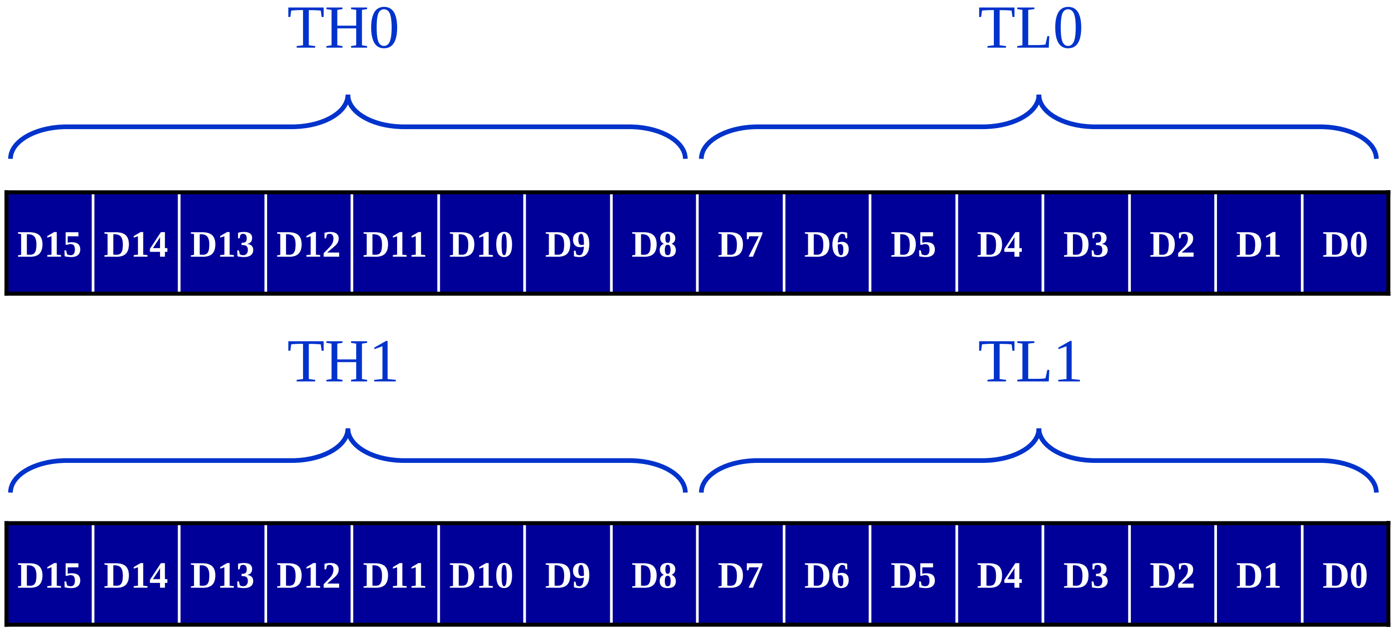
PROGRAMMING TIMERS

- The 8051 has two timers/counters, they can be used either as
  - Timers to generate a time delay or as
  - Event counters to count events happening outside the microcontroller
- Both Timer 0 and Timer 1 are 16 bits wide
  - Since 8051 has an 8-bit architecture, each 16-bits timer is accessed as two separate registers of low byte and high byte

PROGRAMMING TIMERS

Timer 0 & 1 Registers

- ❑ Accessed as low byte and high byte
  - ➢ The low byte register is called TL0/TL1 and
  - ➢ The high byte register is called TH0/TH1
  - ➢ Accessed like any other register
    - ▪ `MOV TL0,#4FH`
    - ▪ `MOV R5,TH0`



TH0 TL0

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

TH1 TL1

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

PROGRAMMING TIMERS

TMOD Register

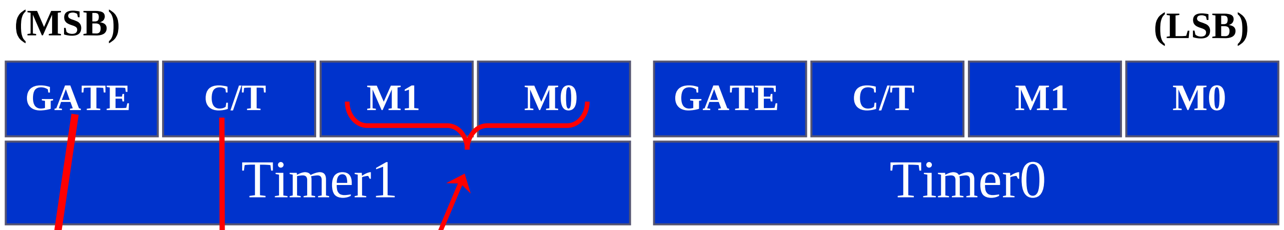
- Both timers 0 and 1 use the same register, called TMOD (timer mode), to set the various timer operation modes
- TMOD is a 8-bit register
  - The lower 4 bits are for Timer 0
  - The upper 4 bits are for Timer 1
  - In each case,
    - The lower 2 bits are used to set the timer mode
    - The upper 2 bits to specify the operation

(MSB) **(LSB)**

GATE	**C/T**	**M1**	**M0**	**GATE**	**C/T**	**M1**	**M0**
Timer1				Timer0			

PROGRAMMING TIMERS

TMOD Register (cont')



(MSB)							(LSB)
GATE	C/T	M1	M0	GATE	C/T	M1	M0
Timer1				Timer0			

M1	M0	Mode	Operating Mode
0	0	0	**13-bit timer mode** 8-bit timer/counter THx with TLx as 5-bit prescaler
0	1	1	**16-bit timer mode** 16-bit timer/counter THx and TLx are cascaded; there is no prescaler
1	0	2	**8-bit auto reload** 8-bit auto reload timer/counter; THx holds a value which is to be reloaded TLx each time it overfolws
1	1	3	**Split timer mode**

Gating control when set. Timer/counter is enable only while the INTx pin is high and the TRx control pin is set

When cleared, the timer is enabled whenever the TRx control bit is set

Timer or counter selected

Cleared for timer operation (input from internal system clock)

Set for counter operation (input from Tx input pin)

PROGRAMMING TIMERS

TMOD Register (cont')

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

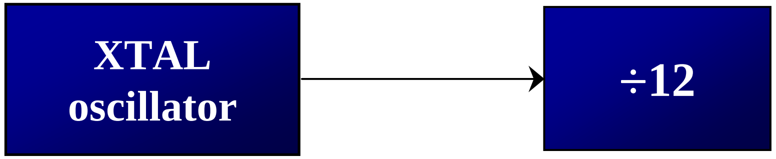
Solution:

We convert the value from hex to binary. From Figure 9-3 we have:
(a) TMOD = 00000001, mode 1 of timer 0 is selected.
(b) TMOD = 00100000, mode 2 of timer 1 is selected.
(c) TMOD = 00010010, mode 2 of timer 0, and mode 1 of timer 1 are selected.

If C/T = 0, it is used as a timer for time delay generation. The clock source for the time delay is the crystal frequency of the 8051

Find the timer's clock frequency and its period for various 8051-based system, with the crystal frequency 11.0592 MHz when C/T bit of TMOD is 0.

Solution:



1/12 × 11.0529 MHz = 921.6 kHz;
T = 1/921.6 kHz = 1.085 us

PROGRAMMING TIMERS

TMOD Register

GATE

- Timers of 8051 do starting and stopping by either software or hardware control
  - In using software to start and stop the timer where GATE=0
    - The start and stop of the timer are controlled by way of software by the TR (timer start) bits TR0 and TR1
      - The SETB instruction starts it, and it is stopped by the CLR instruction
      - These instructions start and stop the timers as long as GATE=0 in the TMOD register
  - The hardware way of starting and stopping the timer by an external source is achieved by making GATE=1 in the TMOD register

- Timer 0, mode 2
- C/T = 0 to use XTAL clock source
- gate = 0 to use internal (software) start and stop method.

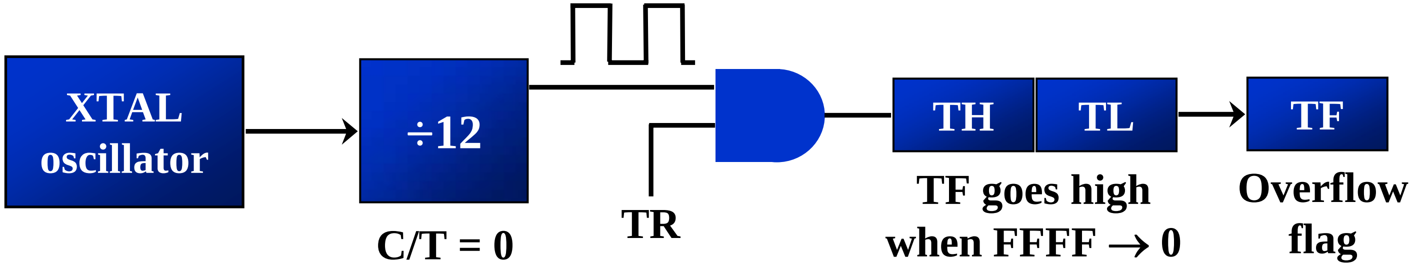
Find the value for TMOD if we want to program timer 0 in mode 2, use 8051 XTAL for the clock source, and use instructions to start and stop the timer.

TMOD = 0000 0010

PROGRAMMING TIMERS

Mode 1 Programming

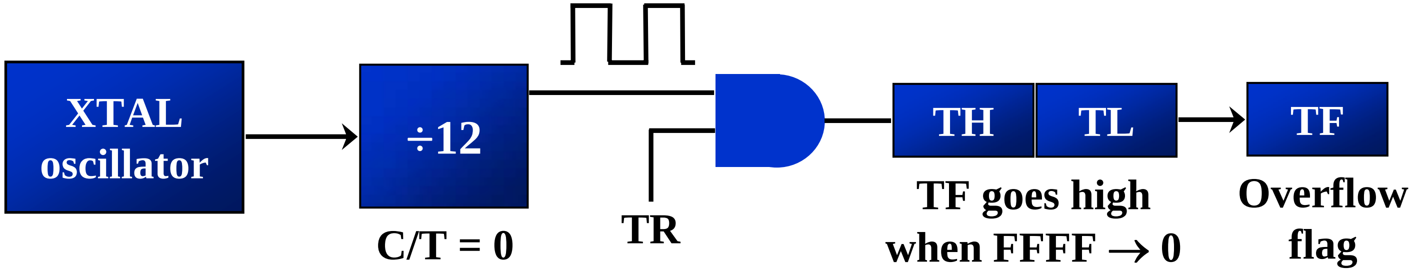
- The following are the characteristics and operations of mode1:
  1. It is a 16-bit timer; therefore, it allows value of 0000 to FFFFH to be loaded into the timer's register TL and TH
  2. After TH and TL are loaded with a 16-bit initial value, the timer must be started
     - This is done by `SETB 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 FFFFH



PROGRAMMING TIMERS

Mode 1 Programming (cont')

3. (cont')
   - When it rolls over from FFFFH to 0000, it sets high a flag bit called TF (timer flag)
     - Each timer has its own timer flag: TF0 for timer 0, and TF1 for timer 1
     - This timer flag can be monitored
   - When this timer flag is raised, one option would be to stop the timer with the instructions `CLR TR0` or `CLR TR1`, for timer 0 and timer 1, respectively
4. After the timer reaches its limit and rolls over, in order to repeat the process
   - TH and TL must be reloaded with the original value, and
   - TF must be reloaded to 0



PROGRAMMING TIMERS

Mode 1 Programming

Steps to Mode 1 Program

- ❑ To generate a time delay
  1. Load the TMOD value register indicating which timer (timer 0 or timer 1) is to be used and which timer mode (0 or 1) is selected
  2. Load registers TL and TH with initial count value
  3. Start the timer
  4. Keep monitoring the timer flag (TF) with the `JNB TFx,target` instruction to see if it is raised
     - Get out of the loop when TF becomes high
  5. Stop the timer
  6. Clear the TF flag for the next round
  7. Go back to Step 2 to load TH and TL again

PROGRAMMING TIMERS

Mode 1 Programming

Finding the Loaded Timer Values

- To calculate the values to be loaded into the TL and TH registers, look at the following example
  - Assume XTAL = 11.0592 MHz, we can use the following steps for finding the TH, TL registers' values
    1. Divide the desired time delay by 1.085 us
    2. Perform 65536 – n, where n is the decimal value we got in Step1
    3. Convert the result of Step2 to hex, where yyxx is the initial hex value to be loaded into the timer's register
    4. Set TL = xx and TH = yy

PROGRAMMING TIMERS

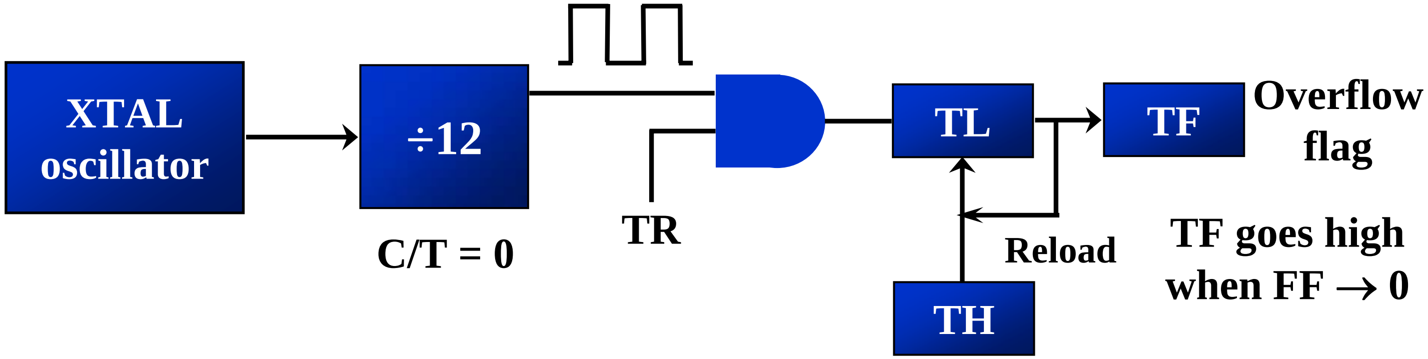
Mode 2 Programming

- The following are the characteristics and operations of mode 2:
  1. It is an 8-bit timer; therefore, it allows only values of 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 of it to TL
     - Then the timer must be 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 register
     - 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)

PROGRAMMING TIMERS

Mode 2 Programming (cont')

4. When the TL register rolls from FFH to 0 and TF is set to 1, TL is reloaded automatically with the original value kept by the TH register
   - To repeat the process, we must simply clear TF and let it go without any need by the programmer to reload the original value
   - This makes mode 2 an auto-reload, in contrast with mode 1 in which the programmer has to reload TH and TL



PROGRAMMING TIMERS

Mode 2 Programming

Steps to Mode 2 Program

- To generate a time delay
  1. Load the TMOD value register indicating which timer (timer 0 or timer 1) is to be used, and the timer mode (mode 2) is selected
  2. Load the TH registers with the initial count value
  3. Start timer
  4. Keep monitoring the timer flag (TF) with the `JNB TFx,target` instruction to see whether it is raised
     - Get out of the loop when TF goes high
  5. Clear the TF flag
  6. Go back to Step4, since mode 2 is auto-reload

COUNTER PROGRAMMING

- Timers can also be used as counters counting events happening outside the 8051
  - When it is used as a counter, it is a pulse outside of the 8051 that increments the TH, TL registers
  - TMOD and TH, TL registers are the same as for the timer discussed previously
- Programming the timer in the last section also applies to programming it as a counter
  - Except the source of the frequency

COUNTER PROGRAMMING

C/T Bit in TMOD Register

- The C/T bit in the TMOD registers decides the source of the clock for the timer
  - When C/T = 1, the timer is used as a counter and gets its pulses from outside the 8051
    - The counter counts up as pulses are fed from pins 14 and 15, these pins are called T0 (timer 0 input) and T1 (timer 1 input)

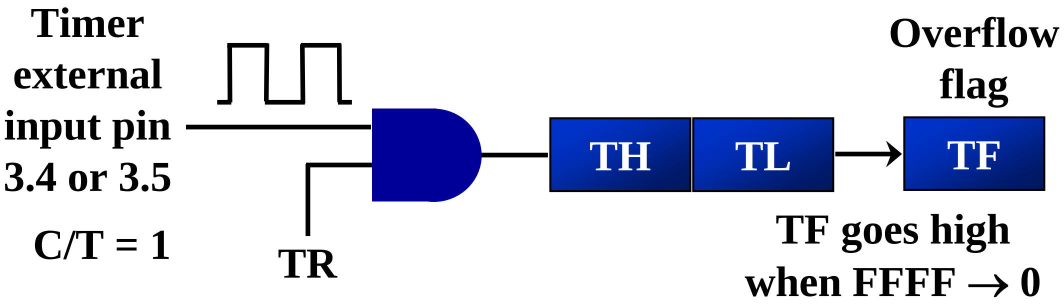
Port 3 pins used for Timers 0 and 1

Pin	Port Pin	Function	Description
14	P3.4	T0	Timer/counter 0 external input
15	P3.5	T1	Timer/counter 1 external input

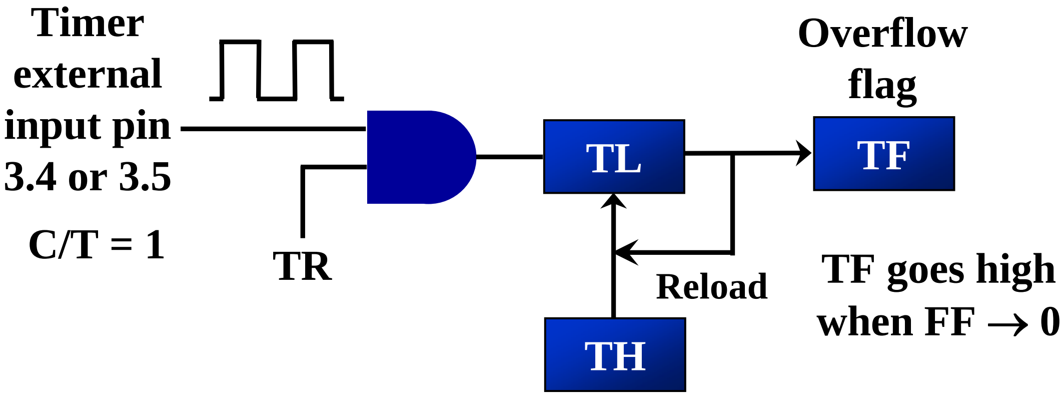
COUNTER PROGRAMMING

C/T Bit in TMOD Register (cont')

Timer with external input (Mode 1)


Timer external input pin 3.4 or 3.5
C/T = 1
TR
TH
TL
TF
Overflow flag
TF goes high when FFFF → 0

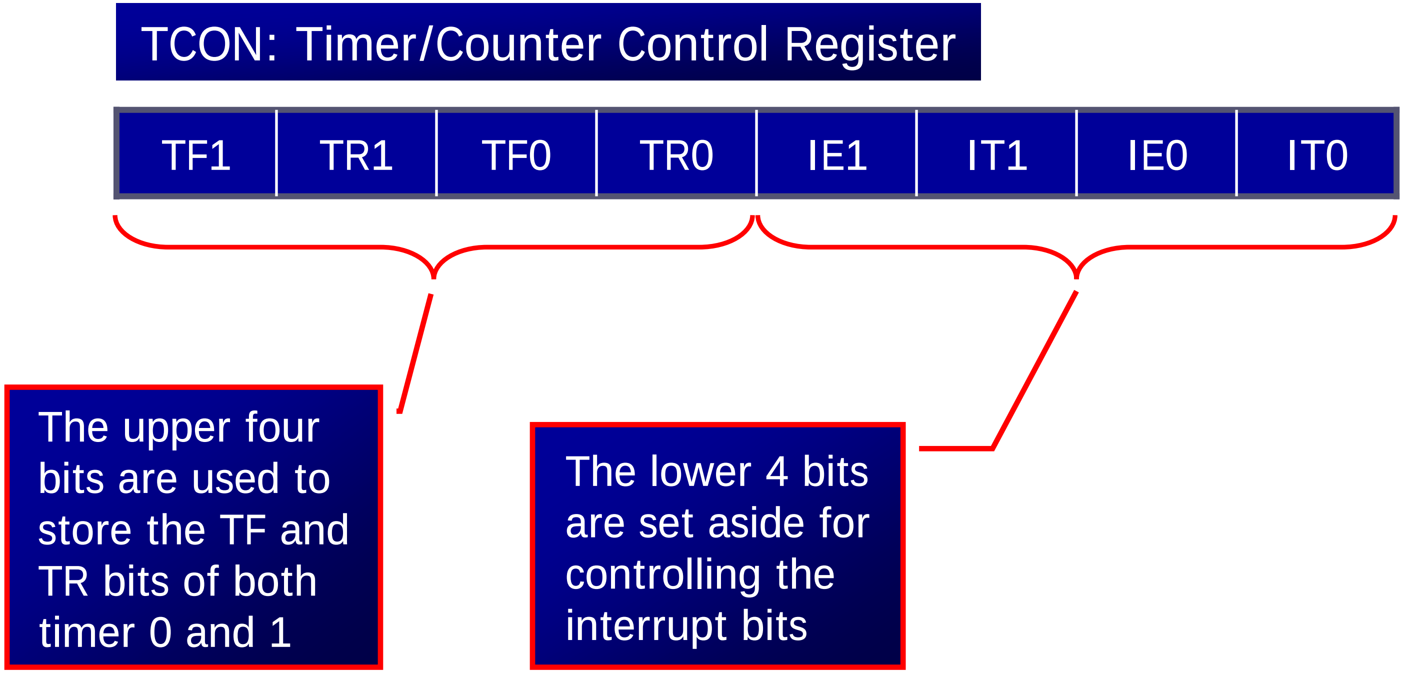
Timer with external input (Mode 2)



COUNTER PROGRAMMING

TCON Register

- TCON (timer control) register is an 8-bit register



TCON: Timer/Counter Control Register

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

The upper four bits are used to store the TF and TR bits of both timer 0 and 1

The lower 4 bits are set aside for controlling the interrupt bits

COUNTER PROGRAMMING

TCON Register (cont')

- TCON register is a bit-addressable register

Equivalent instruction for the Timer Control Register

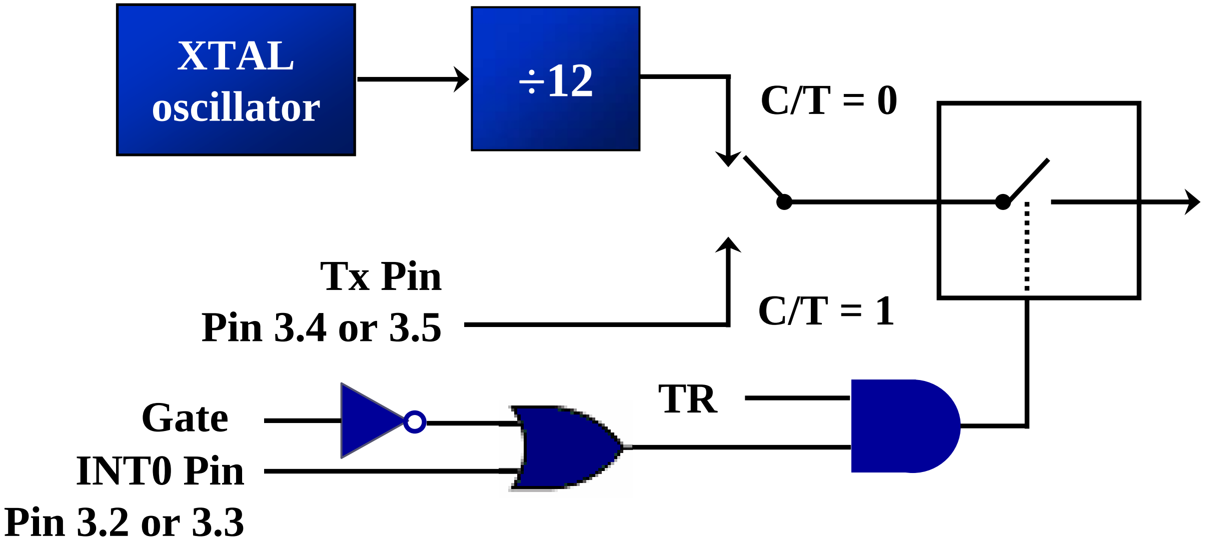
For timer 0		
SETB TR0	=	SETB TCON.4
CLR TR0	=	CLR TCON.4
SETB TF0	=	SETB TCON.5
CLR TF0	=	CLR TCON.5
For timer 1		
SETB TR1	=	SETB TCON.6
CLR TR1	=	CLR TCON.6
SETB TF1	=	SETB TCON.7
CLR TF1	=	CLR TCON.7

COUNTER PROGRAMMING

TCON Register

Case of GATE = 1

- If GATE = 1, the start and stop of the timer are done externally through pins P3.2 and P3.3 for timers 0 and 1, respectively
  - This hardware way allows to start or stop the timer externally at any time via a simple switch


XTAL oscillator
÷12
C/T = 0
Tx Pin
Pin 3.4 or 3.5
C/T = 1
TR
Gate
INT0 Pin
Pin 3.2 or 3.3