OBJECTIVES

- List the timers of the 8051 and their associated registers
- Describe the various modes of the 8051 timers
- Program the 8051 timers in Assembly to generate time delay

PROGRAMMING 8051 TIMERS

- **Basic registers of the timer**
  - **Timer 0 and Timer 1 are 16 bits wide**
  - **each 16-bit timer is accessed as two separate registers of low byte and high byte.**

PROGRAMMING 8051 TIMERS

- **Timer 0 registers**
  - **low byte register is called TL0 (Timer 0 low byte) and the high byte register is referred to as TH0 (Timer 0 high byte)**
  - **can be accessed like any other register, such as A, B, R0, R1, R2, etc.**
  - **"MOV TL0, #4 FH" moves the value 4FH into TL0**
  - **"MOV R5, TH0" saves TH0 (high byte of Timer 0) in R5**

PROGRAMMING 8051 TIMERS

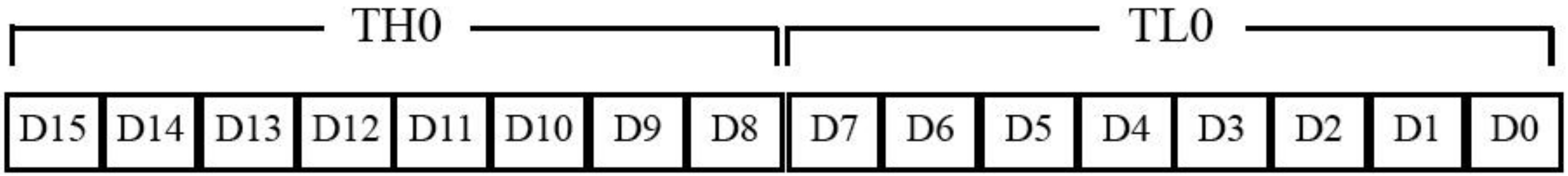
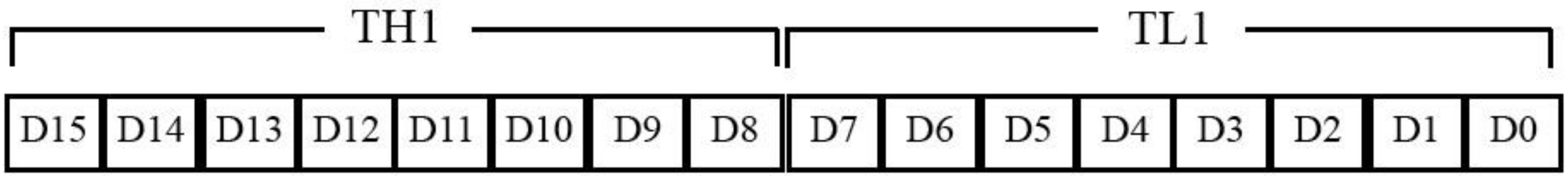


Figure 9–1 Timer 0 Registers

PROGRAMMING 8051 TIMERS

- **Timer 1 registers**
  - **also 16 bits**
  - **split into two bytes TL1 (Timer 1 low byte) and TH1 (Timer 1 high byte)**
  - **accessible in the same way as the registers of Timer 0.**

PROGRAMMING 8051 TIMERS



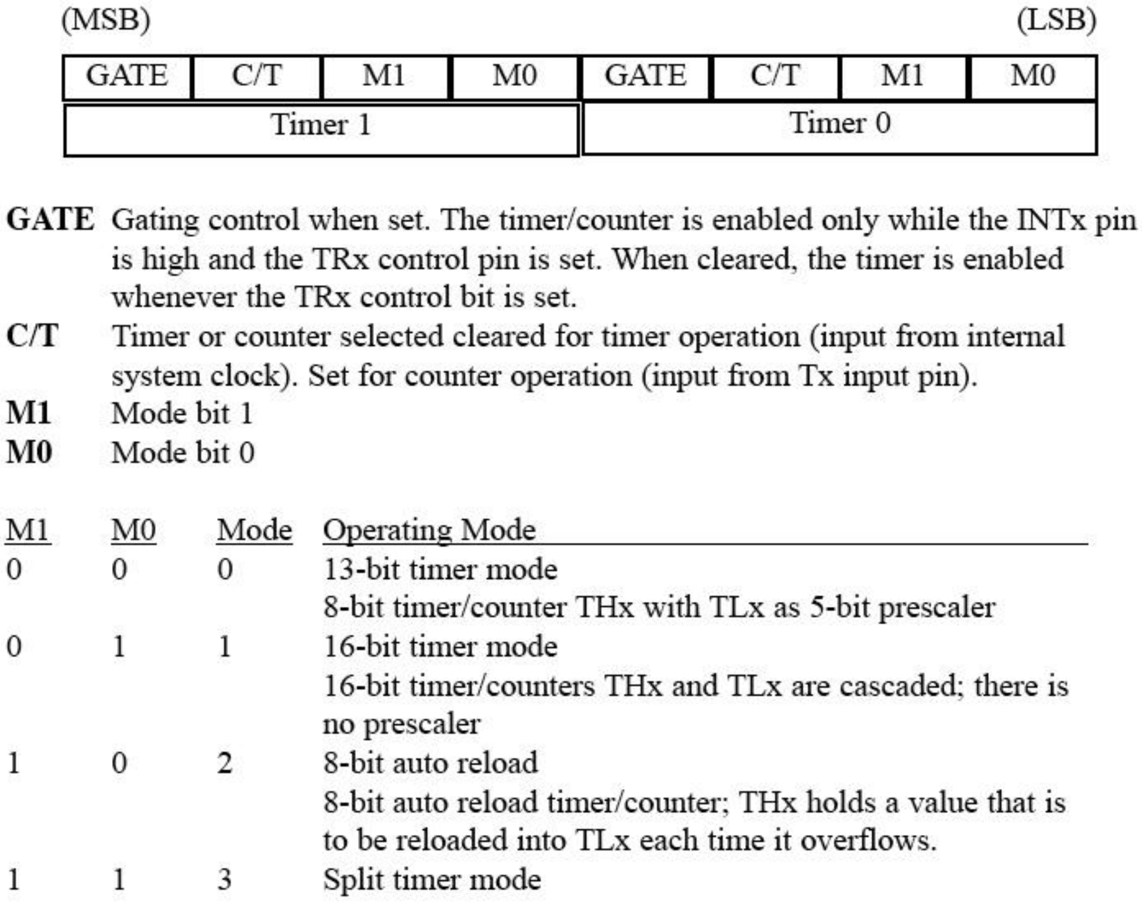
TH1								TL1							
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Figure 9–2 Timer 1 Registers

PROGRAMMING 8051 TIMERS

- **TMOD (timer mode) register**
  - timers 0 and 1 use TMOD register to set operation modes (only learn Mode 1 and 2)
  - 8-bit register
  - lower 4 bits are for Timer 0
  - upper 4 bits are for Timer 1
  - lower 2 bits are used to set the timer mode
    - (only learn Mode 1 and 2)
  - upper 2 bits to specify the operation
    - (only learn timer operation)

PROGRAMMING 8051 TIMERS



(MSB) (LSB)

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

GATE Gating control when set. The timer/counter is enabled 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.

C/T Timer or counter selected cleared for timer operation (input from internal system clock). Set for counter operation (input from Tx input pin).

M1 Mode bit 1

M0 Mode bit 0

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/counters 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 that is to be reloaded into TLx each time it overflows.
1	1	3	Split timer mode

Figure 9–3 TMOD Register

PROGRAMMING 8051 TIMERS

- Clock source for timer
  - timer needs a clock pulse to tick
  - if C/T = 0, the crystal frequency attached to the 8051 is the source of the clock for the timer
  - frequency for the timer is always 1/12th the frequency of the crystal attached to the 8051
  - XTAL = 11.0592 MHz allows the 8051 system to communicate with the PC with no errors
  - In our case, the timer frequency is 1MHz since our crystal frequency is 12MHz

PROGRAMMING 8051 TIMERS

- Mode 1 programming
  - 16-bit timer, values of 0000 to FFFFH
  - TH and TL are loaded with a 16-bit initial value
  - timer started by "SETB TR0" for Timer 0 and "SETB TR1" for Timer l
  - timer count ups until it reaches its limit of FFFFH
  - rolls over from FFFFH to 0000H
  - sets TF (timer flag)
  - when this timer flag is raised, can stop the timer with "CLR TR0" or "CLR TR1“
  - after the timer reaches its limit and rolls over, the registers TH and TL must be reloaded with the original value and TF must be reset to 0

PROGRAMMING 8051 TIMERS

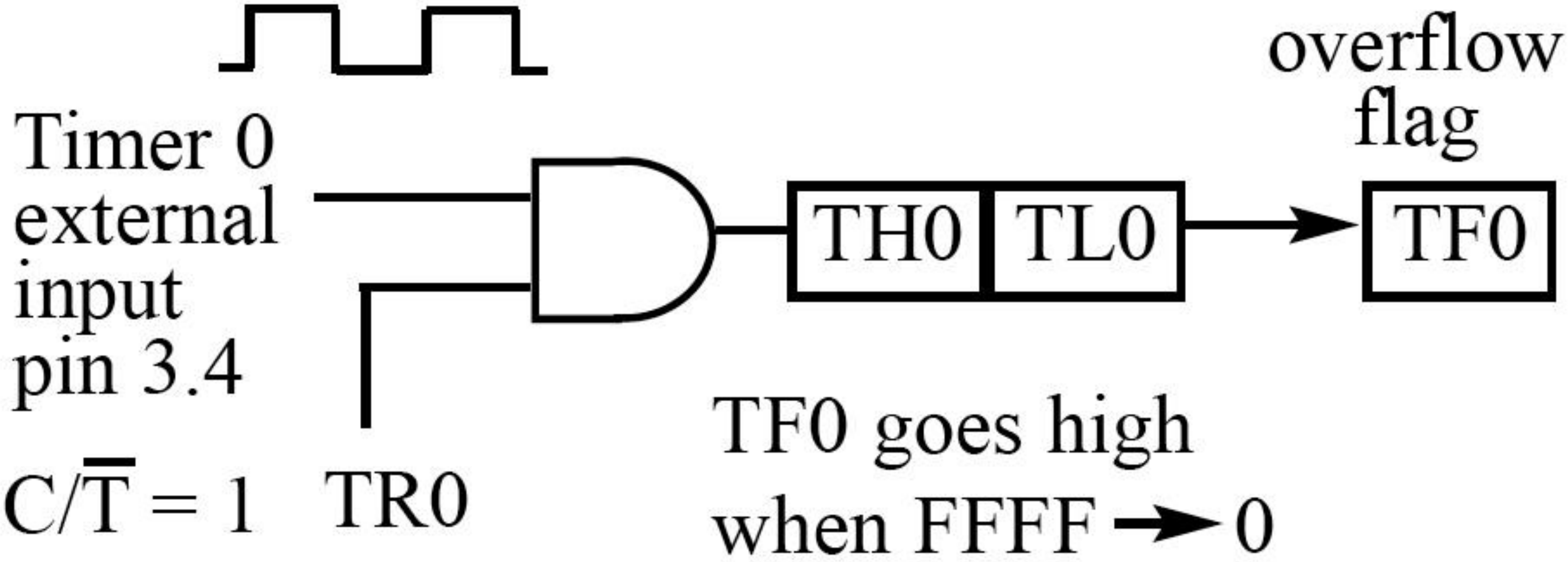


Figure 9–5a Timer 0 with External Input (Mode 1)

PROGRAMMING 8051 TIMERS (for information only)

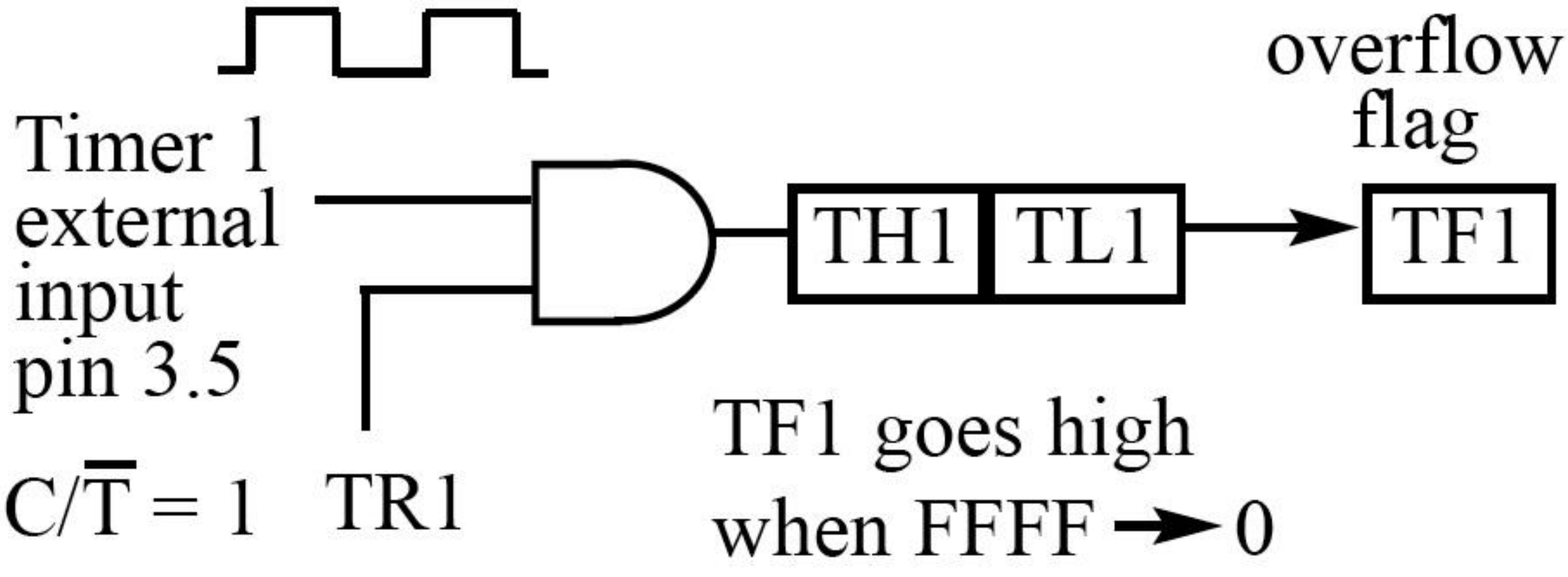


Figure 9–5b Timer 1 with External Input (Mode 1)

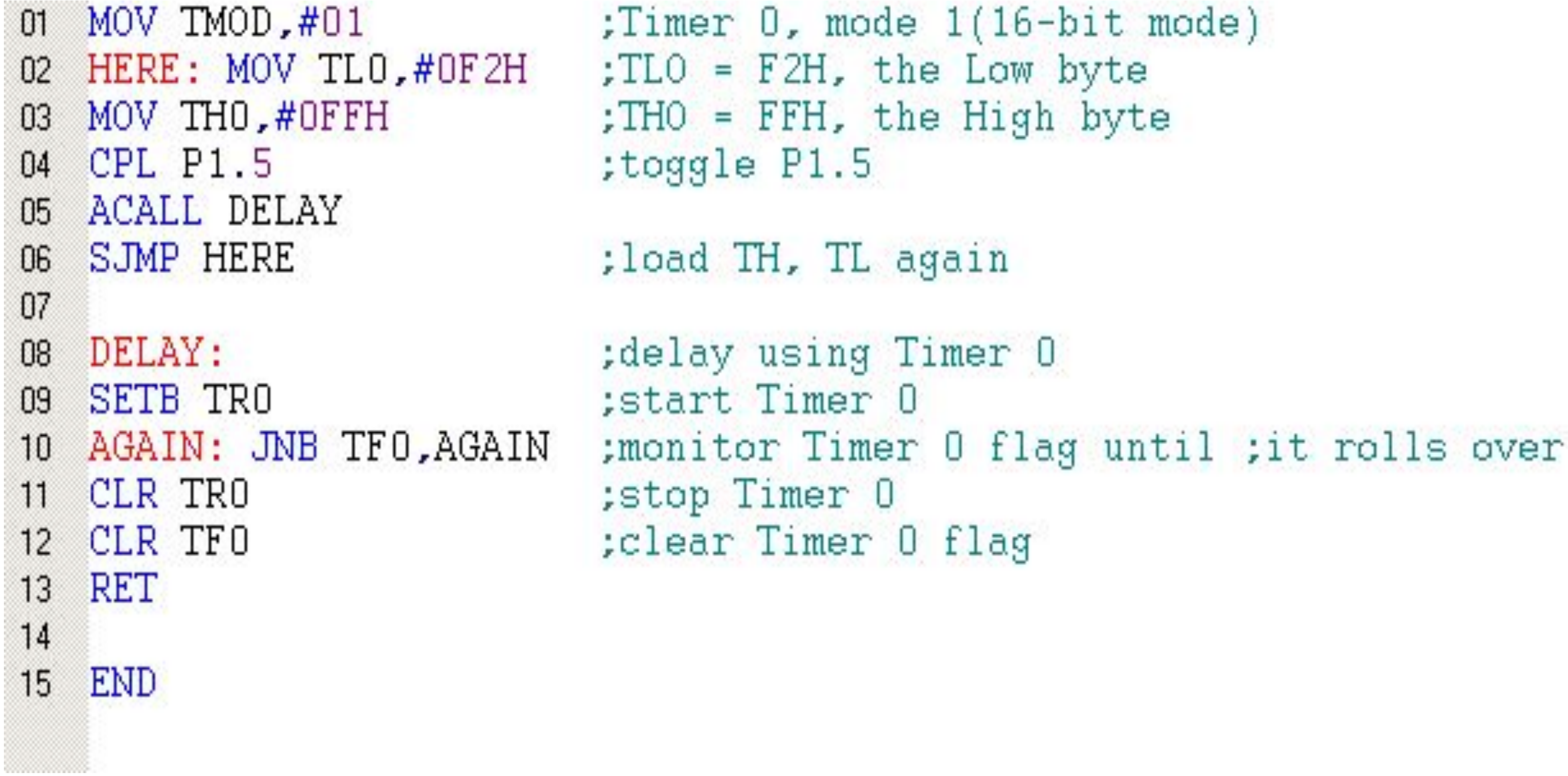
PROGRAMMING 8051 TIMERS

- Steps to program in mode 1
  - Set timer mode 1 or 2
  - Set TL0 and TH0 (for mode 1 16 bit mode)
  - Set TH0 only (for mode 2 8 bit auto reload mode)
  - Run the timer
  - Monitor the timer flag bit

Example

In the following program, we are creating a square wave of 50% duty cycle (with equal portions high and low) on the P1.5 bit.

Timer 0 is used to generate the time delay



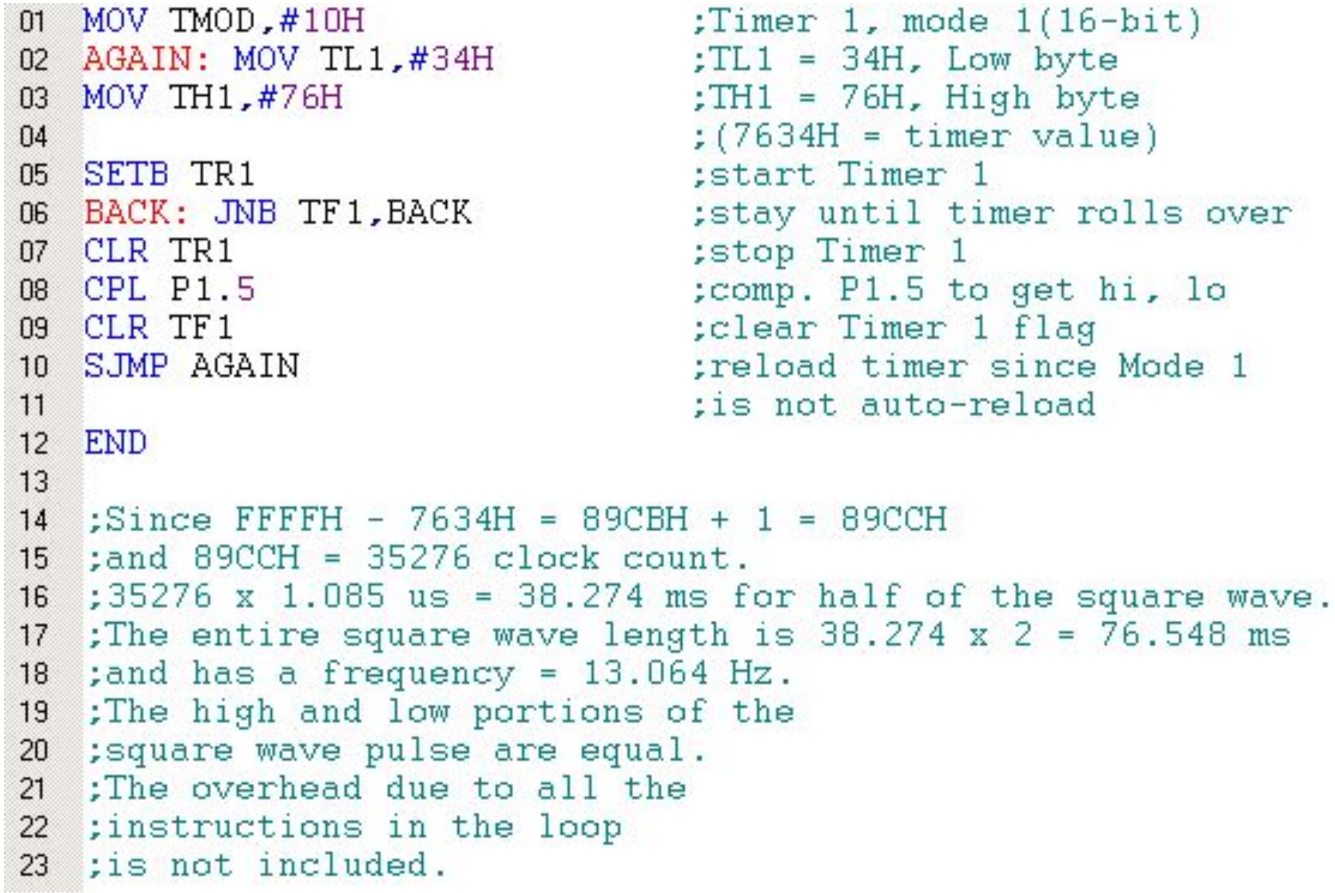
```
MOV TMOD,#01              ;Timer 0, mode 1(16-bit mode)
HERE: MOV TL0,#0F2H       ;TL0 = F2H, the Low byte
MOV TH0,#0FFH             ;TH0 = FFH, the High byte
CPL P1.5                  ;toggle P1.5
ACALL DELAY
SJMP HERE                 ;load TH, TL again

DELAY:                    ;delay using Timer 0
SETB TR0                  ;start Timer 0
AGAIN: JNB TF0,AGAIN      ;monitor Timer 0 flag until ;it rolls over
CLR TR0                   ;stop Timer 0
CLR TF0                   ;clear Timer 0 flag
RET

END
```

Example

The following program generates a square wave on pin P 1.5 continuously using Timer 1 for a time delay. Find the frequency of the square wave if XTAL = 11.0592 MHz. In your calculation do not include the overhead due to the timer setup instructions in the loop.



```
MOV TMOD,#10H               ;Timer 1, mode 1(16-bit)
AGAIN: MOV TL1,#34H         ;TL1 = 34H, Low byte
MOV TH1,#76H                ;TH1 = 76H, High byte
                            ;(7634H = timer value)
SETB TR1                    ;start Timer 1
BACK: JNB TF1,BACK          ;stay until timer rolls over
CLR TR1                     ;stop Timer 1
CPL P1.5                    ;comp. P1.5 to get hi, lo
CLR TF1                     ;clear Timer 1 flag
SJMP AGAIN                  ;reload timer since Mode 1
                            ;is not auto-reload
END

;Since FFFFH - 7634H = 89CBH + 1 = 89CCH
;and 89CCH = 35276 clock count.
;35276 x 1.085 us = 38.274 ms for half of the square wave.
;The entire square wave length is 38.274 x 2 = 76.548 ms
;and has a frequency = 13.064 Hz.
;The high and low portions of the
;square wave pulse are equal.
;The overhead due to all the
;instructions in the loop
;is not included.
```

SECTION 9.1: PROGRAMMING 8051 TIMERS

- **Finding values to be loaded into the timer**
  - **XTAL = 11.0592 MHz (12MHz)**
  - **divide the desired time delay by 1.085µs (1µs) to get *n***
  - **65536 – *n* = *N***
  - **convert *N* to hex yyxx**
  - **set TL = xx and TH = yy**

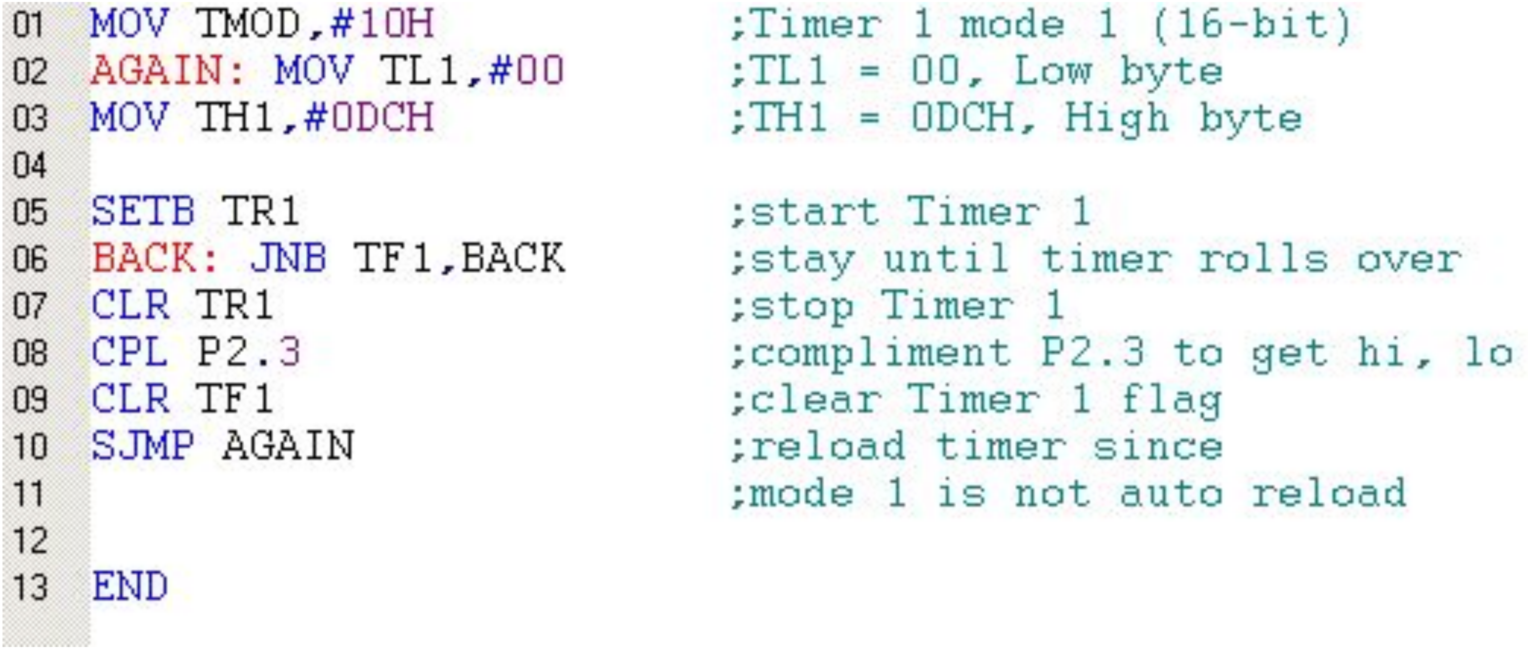
Example

Assuming XTAL = 11.0592 MHz, write a program to generate a square wave of 50 Hz frequency on pin P2.3.

- T = 1/50 Hz = 20 ms
- 1/2 of it for the high and low portions of the pulse = 10 ms
- 10 ms / 1.085 us = 9216
- 65536 - 9216 = 56320 in decimal = DC00H
- TL = 00 and TH = DCH
- The calculation for 12MHz crystal uses the same steps

Example (cont)

Assuming XTAL = 11.0592 MHz, write a program to generate a square wave of 50 Hz frequency on pin P2.3.



```
MOV TMOD,#10H                  ;Timer 1 mode 1 (16-bit)
AGAIN: MOV TL1,#00             ;TL1 = 00, Low byte
MOV TH1,#0DCH                  ;TH1 = 0DCH, High byte

SETB TR1                       ;start Timer 1
BACK: JNB TF1,BACK             ;stay until timer rolls over
CLR TR1                        ;stop Timer 1
CPL P2.3                       ;compliment P2.3 to get hi, lo
CLR TF1                        ;clear Timer 1 flag
SJMP AGAIN                     ;reload timer since
                               ;mode 1 is not auto reload

END
```

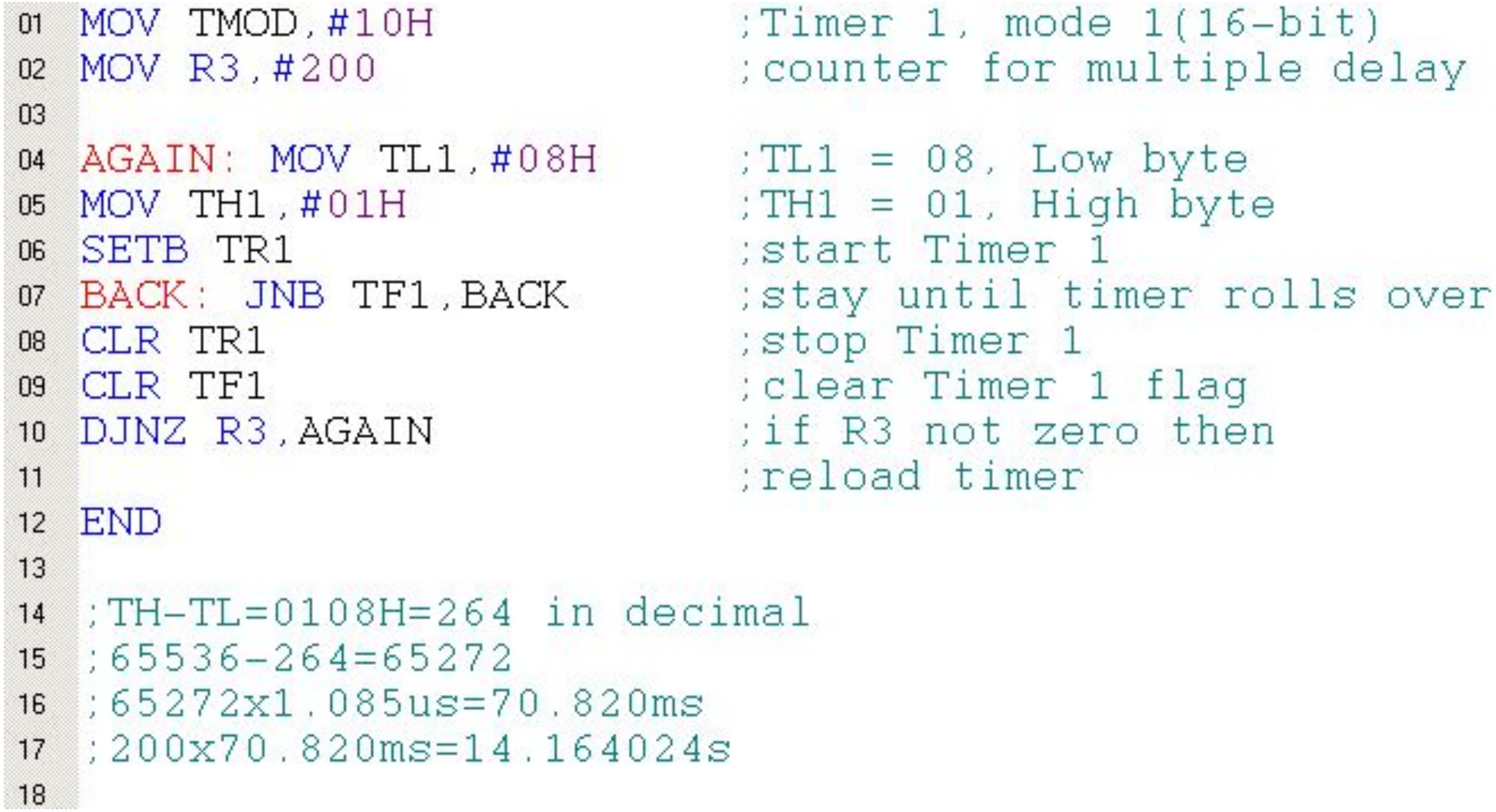
PROGRAMMING 8051 TIMERS

- **Generating a large time delay**
  - **size of the time delay depends**
    - **crystal frequency**
    - **timer's 16-bit register in mode 1**
  - **largest time delay is achieved by making both TH and TL zero**
  - **what if that is not enough?**

- **Using Windows calculator to find TH, TL**
  - Windows scientific calculator can be use to find the TH, TL values
  - Lets say we would like to find the TH, TL values for a time delay that uses 35,000 clocks of 1.085μs
    1. open scientific calculator and select decimal
    2. enter 35,000
    3. select hex - converts 35,000 to hex 88B8H
    4. select +/- to give -35000 decimal (7748H)
    5. the lowest two digits (48) of this hex value are for TL and the next two (77) are for TH

Example

Examine the following program and find the time delay in seconds. Exclude the time delay due to the instructions in the loop.



```
MOV TMOD,#10H                 ;Timer 1, mode 1(16-bit)
MOV R3,#200                   ;counter for multiple delay

AGAIN: MOV TL1,#08H           ;TL1 = 08, Low byte
MOV TH1,#01H                  ;TH1 = 01, High byte
SETB TR1                      ;start Timer 1
BACK: JNB TF1,BACK            ;stay until timer rolls over
CLR TR1                       ;stop Timer 1
CLR TF1                       ;clear Timer 1 flag
DJNZ R3,AGAIN                 ;if R3 not zero then
                              ;reload timer
END

;TH-TL=0108H=264 in decimal
;65536-264=65272
;65272x1.085us=70.820ms
;200x70.820ms=14.164024s

```

PROGRAMMING 8051 TIMERS (for information only)

- **Mode 0**
  - **works like mode 1**
  - **13-bit timer instead of 16bit**
  - **13-bit counter hold values 0000 to 1FFFH**
  - **when the timer reaches its maximum of 1FFFH, it rolls over to 0000, and TF is set**

PROGRAMMING 8051 TIMERS

- **Mode 2 programming**
  - 8-bit timer, allows values of 00 to FFH
  - TH is loaded with the 8-bit value
  - a copy is given to TL
  - timer is started by ,"SETB TR0" or "SETB TR1“
  - starts to count up by incrementing the TL register
  - counts up until it reaches its limit of FFH
  - when it rolls over from FFH to 00, it sets high TF
  - TL is reloaded automatically with the value in TH
  - To repeat, clear TF
  - mode 2 is an auto-reload mode

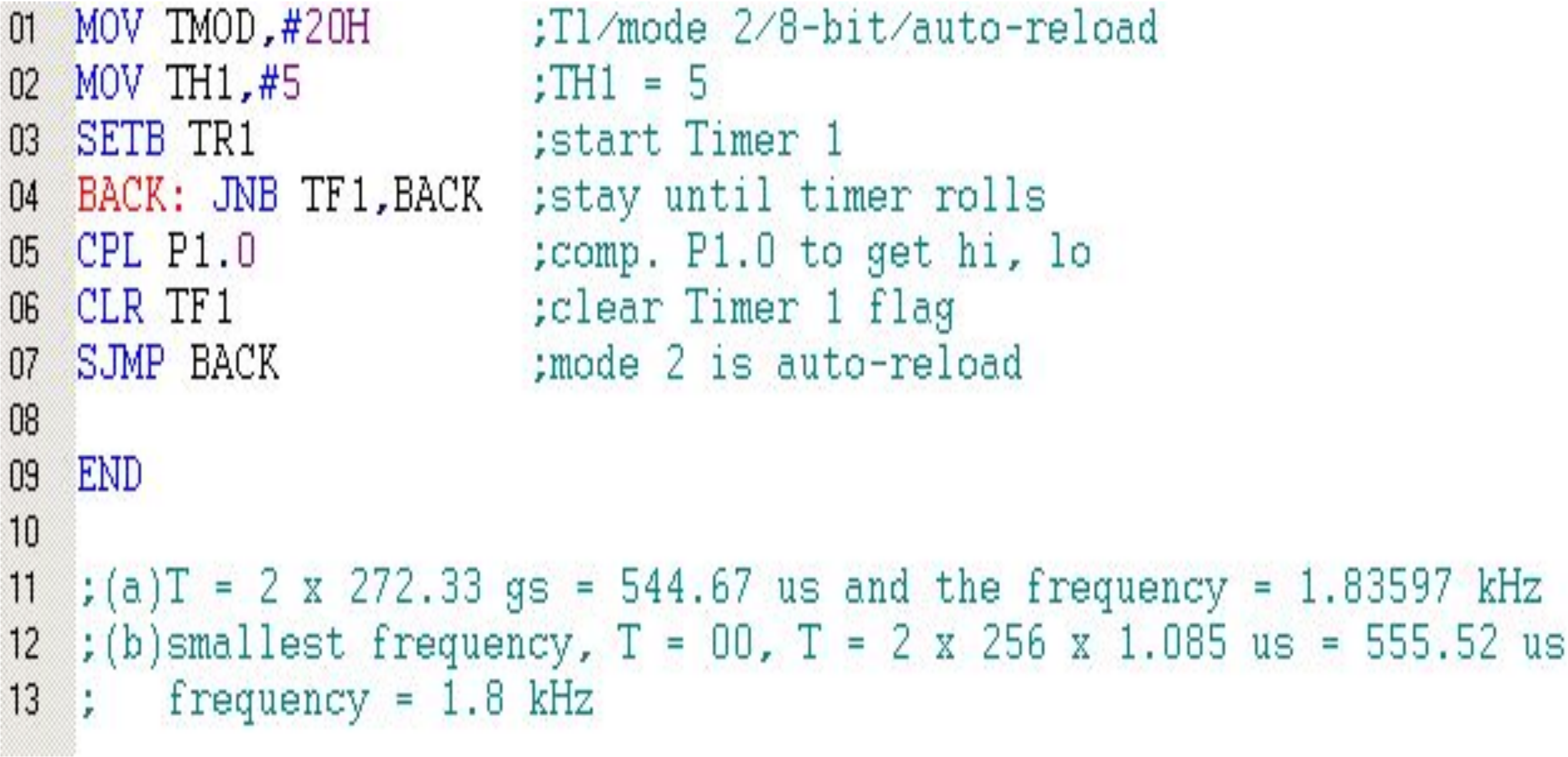
PROGRAMMING 8051 TIMERS

- **Steps to program in mode 2**

1. **load TMOD, select mode 2**
2. **load the TH**
3. **start timer**
4. **monitor the timer flag (TF) with "JNB”**
5. **get out of the loop when TF=1**
6. **clear TF**
7. **go back to Step 4 since mode 2 is auto-reload**

Example

Assuming that XTAL = 11.0592 MHz, find (a) the frequency of the square wave generated on pin P1.0 and (b) the smallest frequency achievable in this program, and the TH value to do that.



```
MOV TMOD,#20H           ;T1/mode 2/8-bit/auto-reload
MOV TH1,#5              ;TH1 = 5
SETB TR1                ;start Timer 1
BACK: JNB TF1,BACK      ;stay until timer rolls
CPL P1.0                ;comp. P1.0 to get hi, lo
CLR TF1                 ;clear Timer 1 flag
SJMP BACK               ;mode 2 is auto-reload

END

;(a)T = 2 x 272.33 gs = 544.67 us and the frequency = 1.83597 kHz
;(b)smallest frequency, T = 00, T = 2 x 256 x 1.085 us = 555.52 us
;    frequency = 1.8 kHz
```

PROGRAMMING 8051 TIMERS

- **Assemblers and negative values**
  - **can let the assembler calculate the value for TH and TL which makes the job easier**
  - **"MOV TH1, # -100", the assembler will calculate the -100 = 9CH**
  - **"MOV TH1,#high(-10000) "**
  - **"MOV TL1,#low(-10000) "**

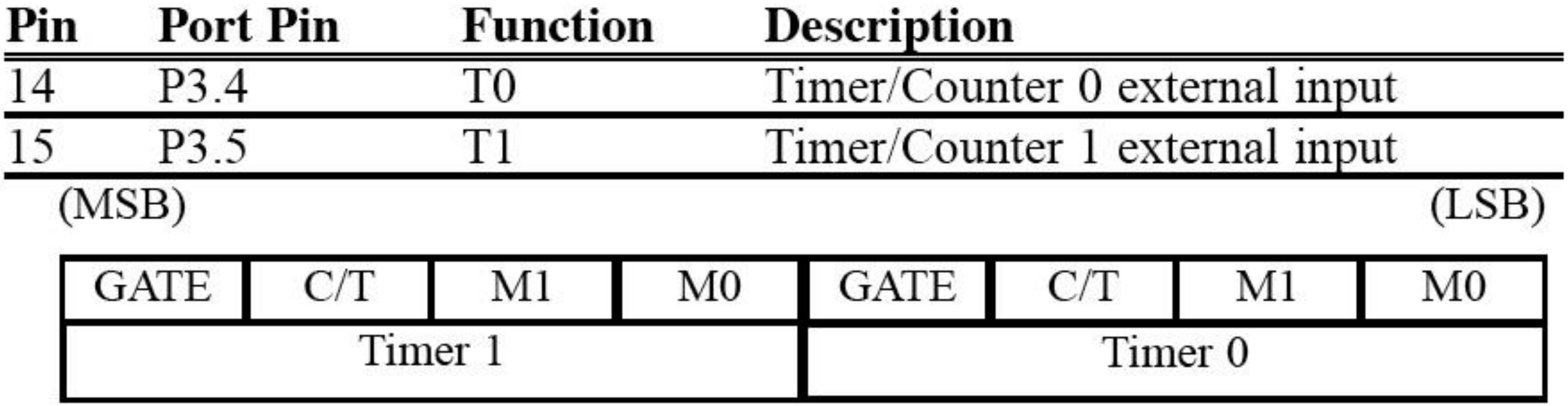
COUNTER PROGRAMMING (for information only)

- C/T bit in TMOD register
  - used as a timer, the 8051's crystal is used as the source of the frequency
  - used as a counter, pulse outside the 8051 increments the TH, TL registers
  - counter mode, TMOD and TH, TL registers are the same as for the timer
  - timer modes are the same as well

COUNTER PROGRAMMING (for information only)

- **C/T bit in TMOD register**
  - C/T bit in the TMOD register decides the source of the clock for the timer
  - C/T = 0, timer gets pulses from crystal
  - C/T = 1, the timer used as counter and gets pulses from outside the 8051
  - C/T = 1, the counter counts up as pulses are fed from pins 14 and 15
  - pins are called T0 (Timer 0 input) and T1 (Timer 1 input)
  - these two pins belong to port 3
  - Timer 0, when C/T = 1, pin P3.4 provides the clock pulse and the counter counts up for each clock pulse coming from that pin
  - Timer 1, when C/T = 1 each clock pulse coming in from pin P3.5 makes the counter count up

COUNTER PROGRAMMING



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

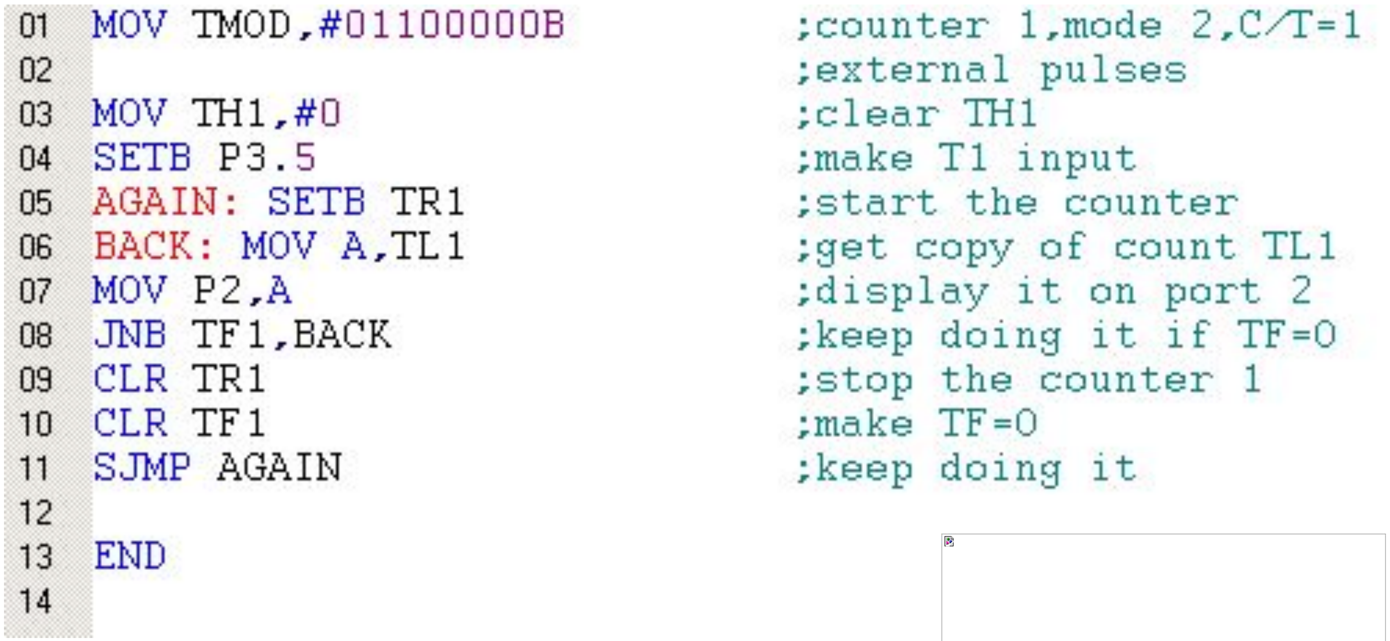
(MSB) (LSB)

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

Table 9–1 Port 3 Pins Used For Timers 0 and 1

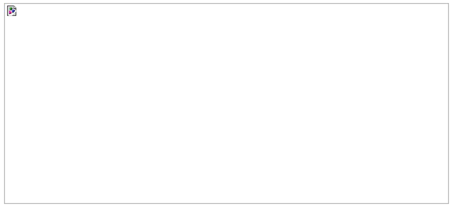
Example 18

Assuming that clock pulses are fed into pin T1, write a program for counter 1 in mode 2 to count the pulses and display the state of the TL1 count on P2. (for information only)



```
MOV TMOD,#01100000B                   ;counter 1,mode 2,C/T=1
                                      ;external pulses
MOV TH1,#0                            ;clear TH1
SETB P3.5                             ;make T1 input
AGAIN: SETB TR1                       ;start the counter
BACK: MOV A,TL1                       ;get copy of count TL1
MOV P2,A                              ;display it on port 2
JNB TF1,BACK                          ;keep doing it if TF=0
CLR TR1                               ;stop the counter 1
CLR TF1                               ;make TF=0
SJMP AGAIN                            ;keep doing it

END
```



to LEDs

P2 is connected to 8 LEDs and input T1 to pulse.

COUNTER PROGRAMMING

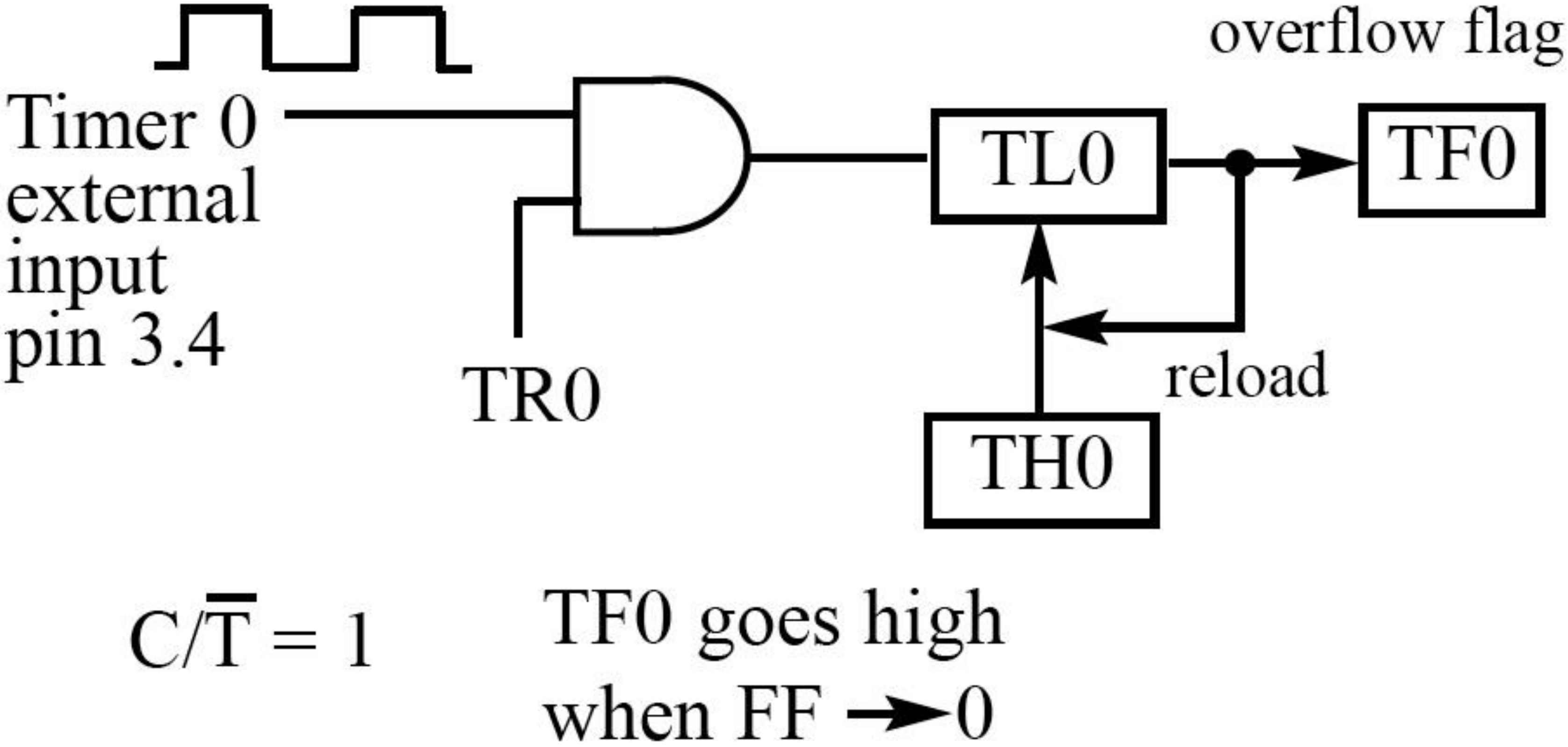


Figure 9–6 Timer 0 with External Input (Mode 2)

COUNTER PROGRAMMING

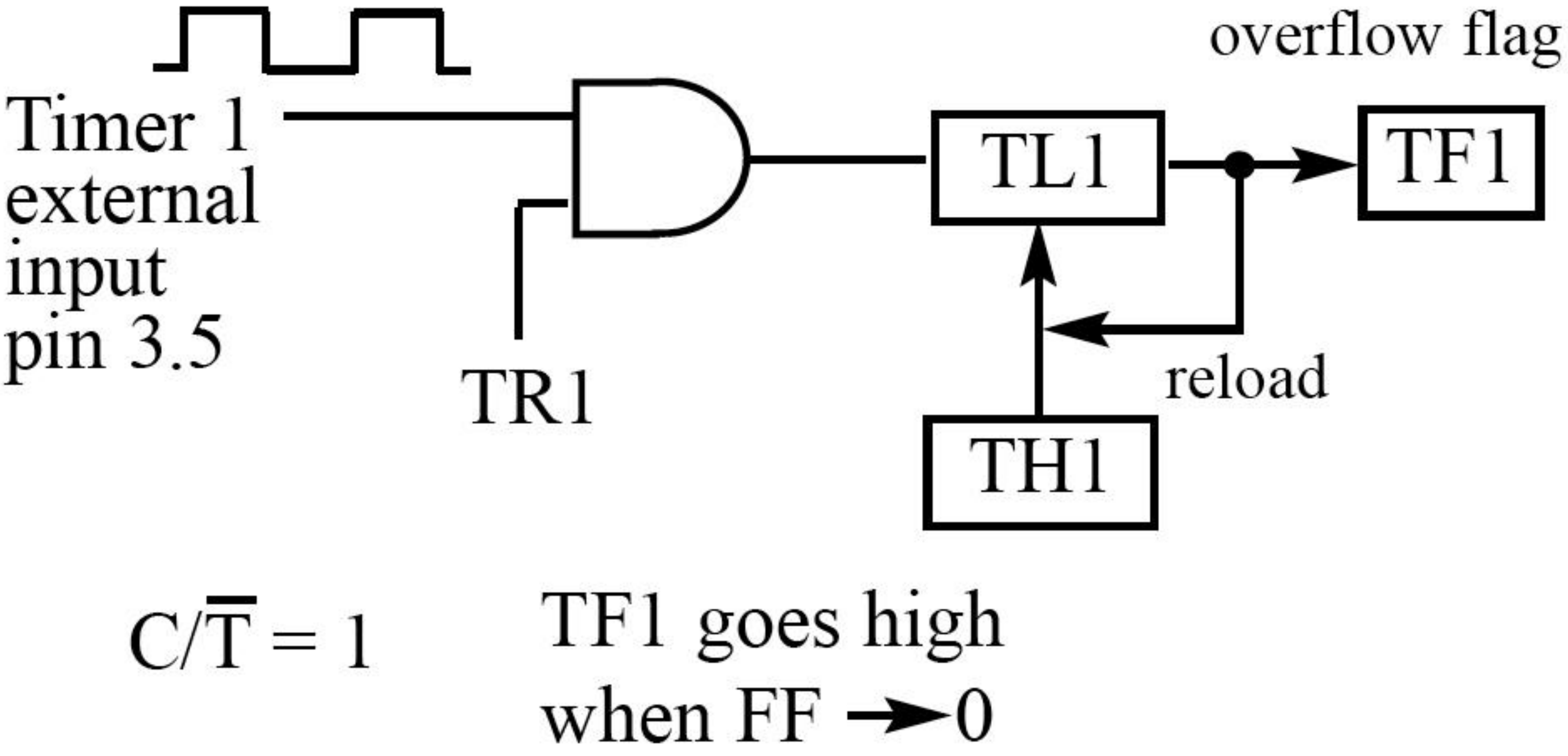
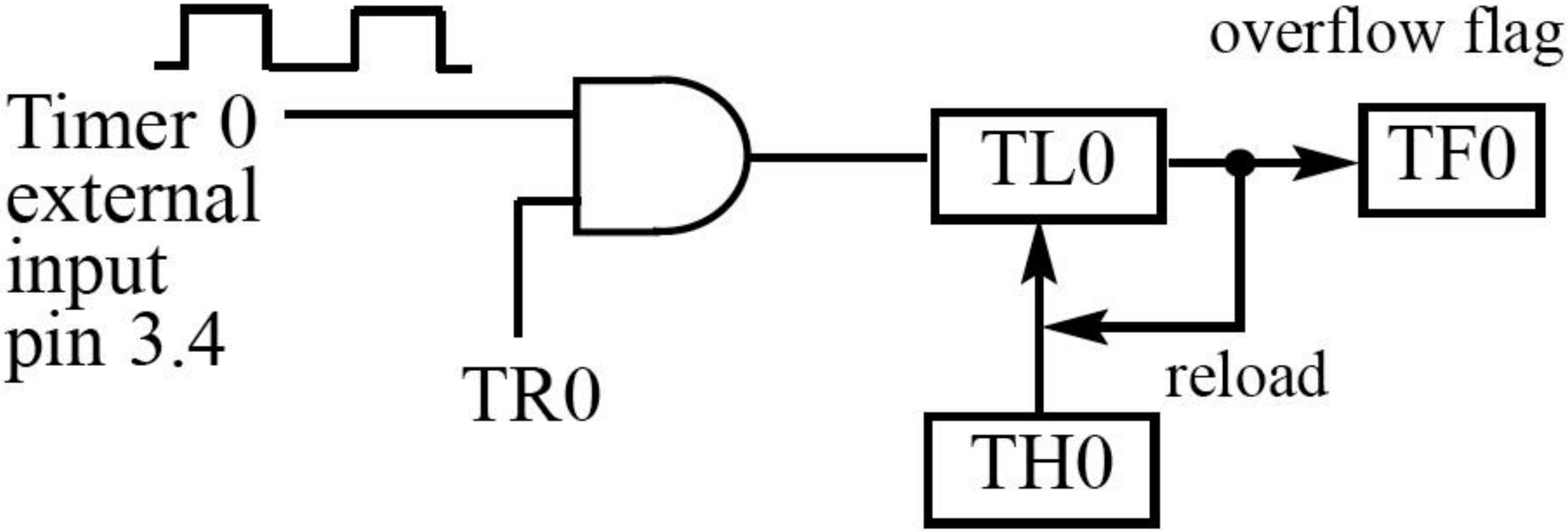


Figure 9–7 Timer 1 with External Input (Mode 2)

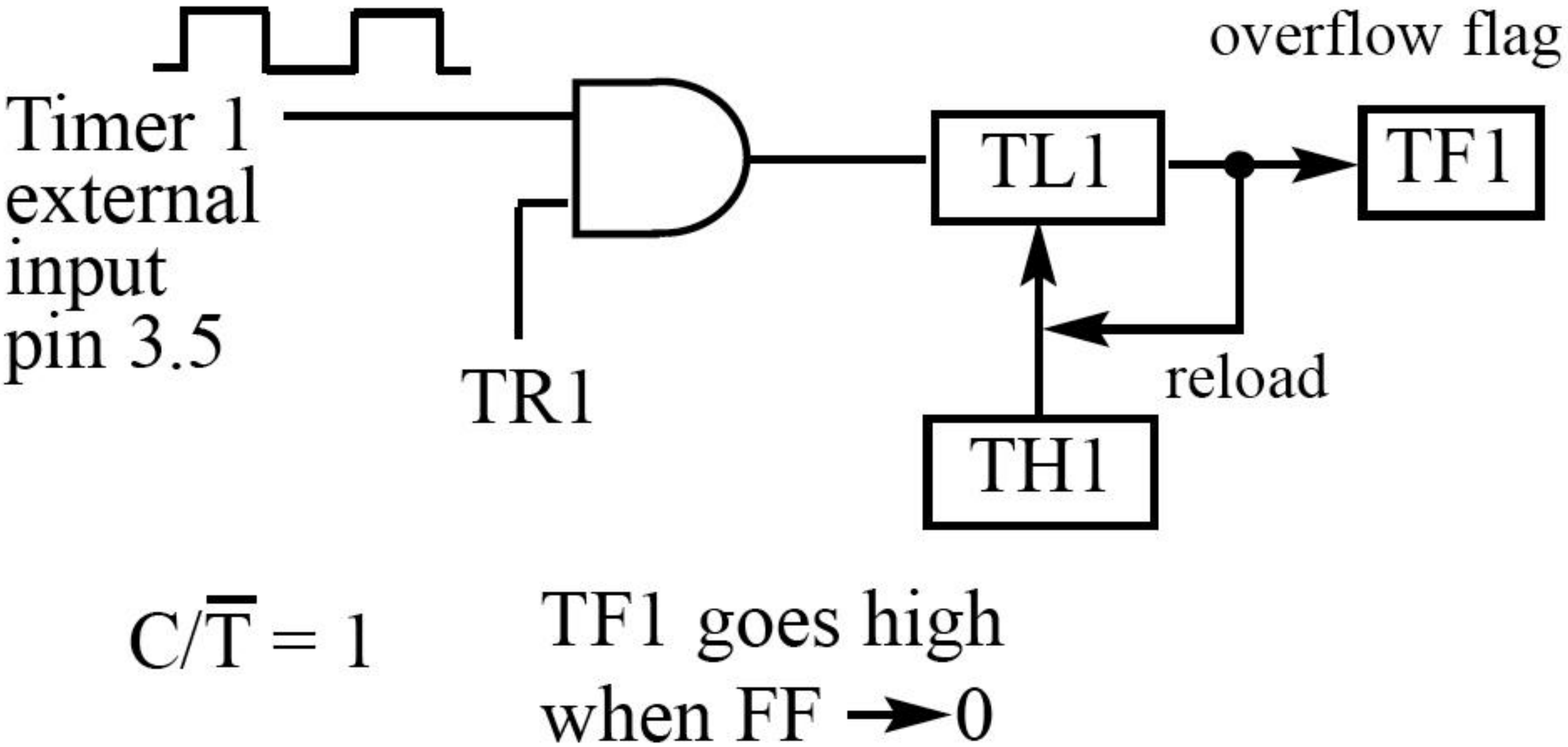
COUNTER PROGRAMMING



$C/\overline{T} = 1$

TF0 goes high
when FF $\rightarrow$ 0

COUNTER PROGRAMMING



COUNTER PROGRAMMING

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

TCON: Timer/Counter Control Register

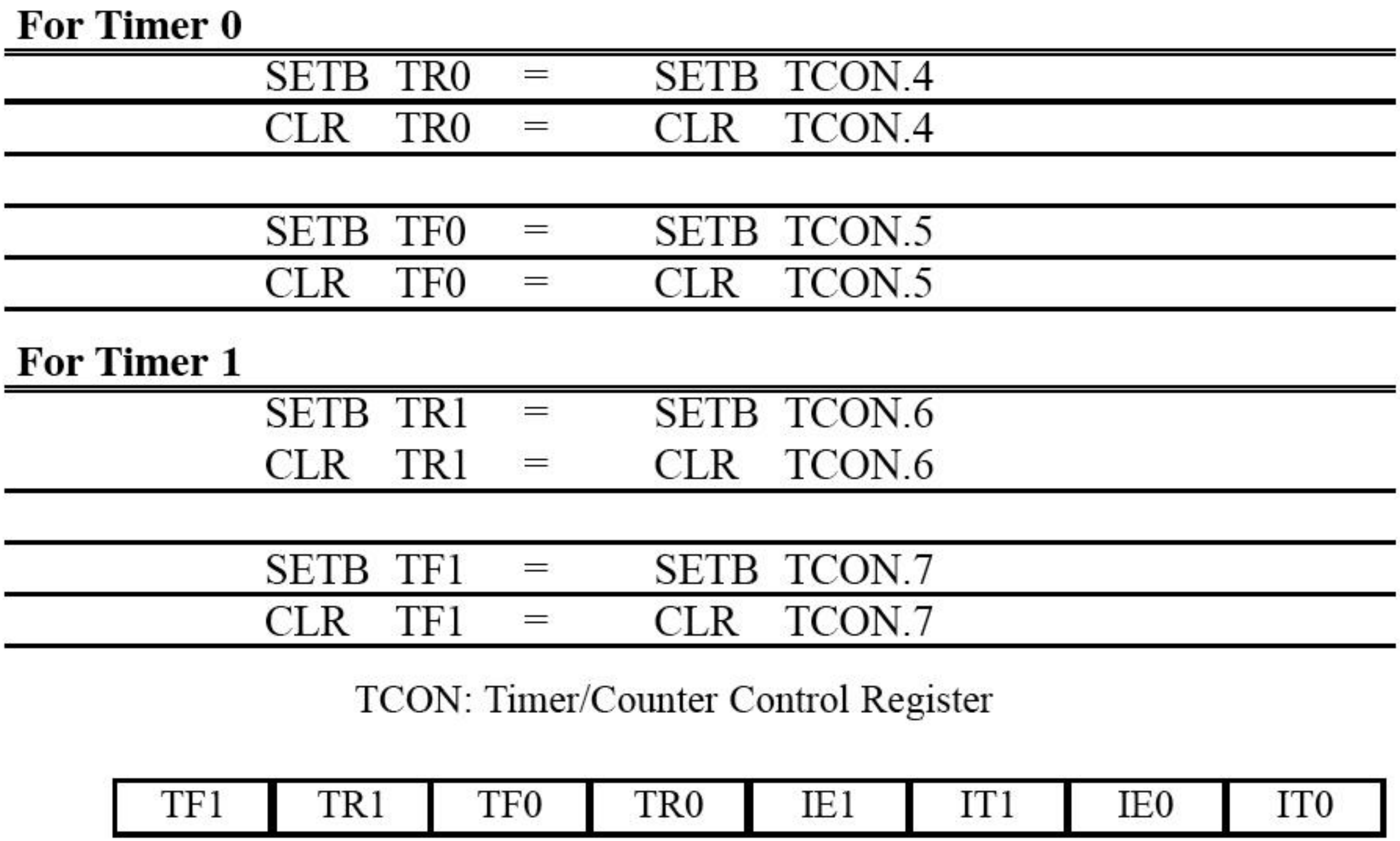
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Table 9–1 Port 3 Pins Used For Timers 0 and 1

COUNTER PROGRAMMING

- **TCON register**
  - **TR0 and TR1 flags turn on or off the timers**
  - **bits are part of a register called TCON (timer control)**
  - **upper four bits are used to store the TF and TR bits of both Timer 0 and Timer 1**
  - **lower four bits are set aside for controlling the interrupt bits**
  - **"SETB TRI" and "CLR TRI“**
  - **"SETB TCON. 6" and "CLR TCON. 6“**

COUNTER PROGRAMMING



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

TCON: Timer/Counter Control Register

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Table 9–2 Equivalent Instructions for the Timer Control Register (TCON)

COUNTER PROGRAMMING

- **The case of GATE = 1 in TMOD**
  - **GATE = 0, the timer is started with instructions** "SETB TR0" and "SETB TR1“
  - **GATE = 1, the start and stop of the timers are done externally through pins P3.2 and P3.3**
  - **allows us to start or stop the timer externally at any time via a simple switch**

COUNTER PROGRAMMING

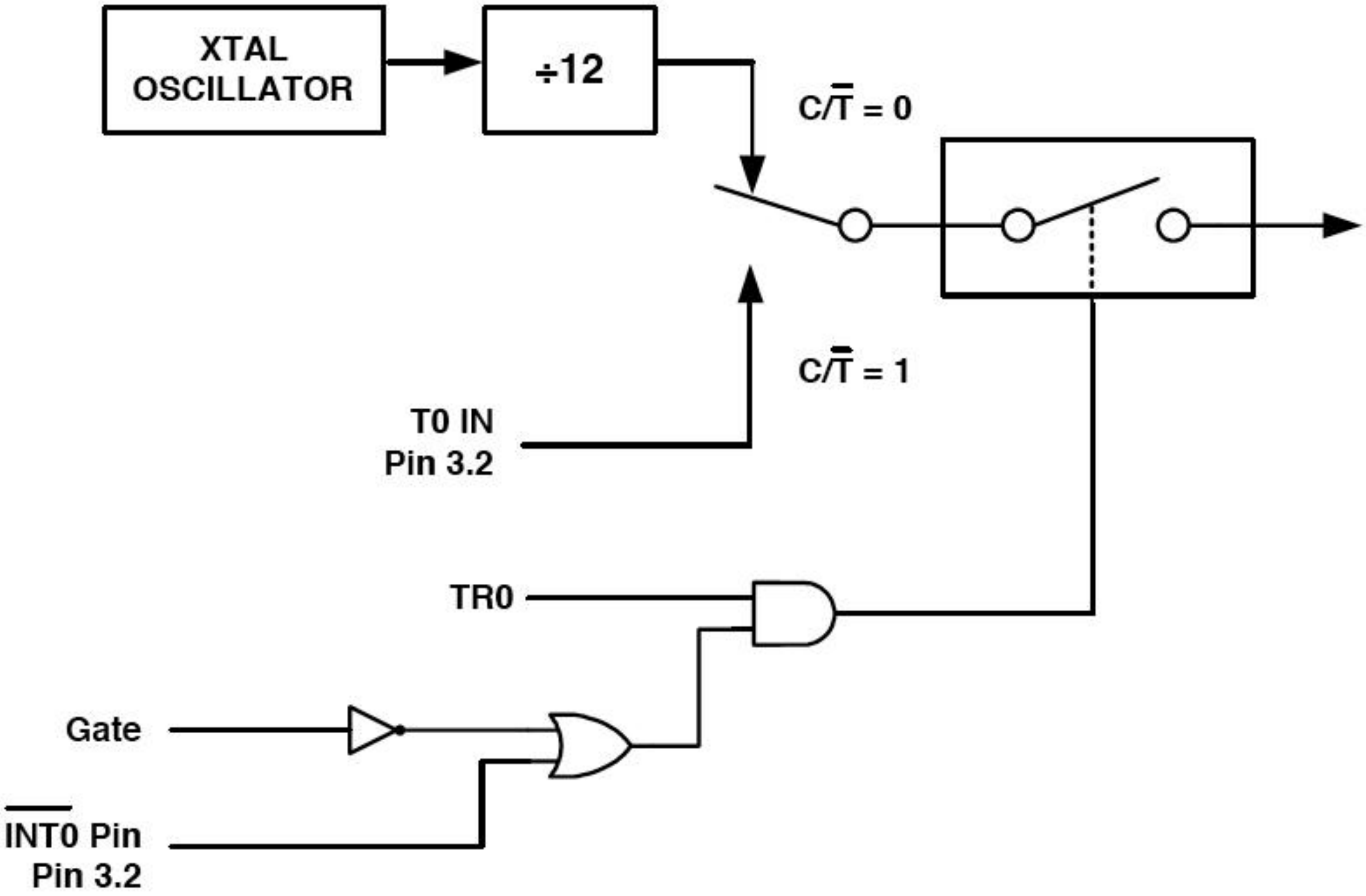


Figure 9–8 Timer/Counter 0

COUNTER PROGRAMMING

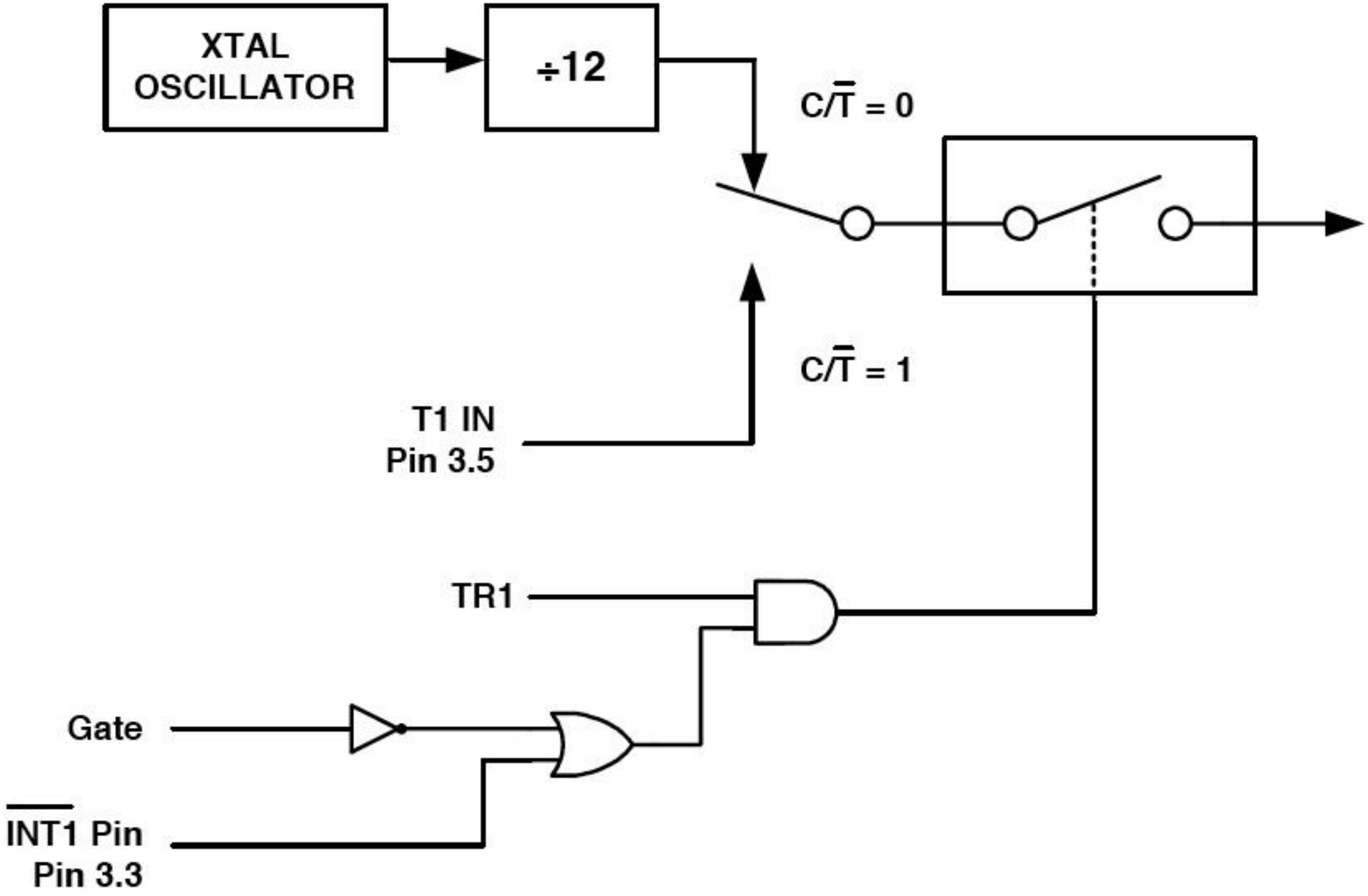


Figure 9–9 Timer/Counter 1

Thank you