The 8051 Microcontroller and Embedded Systems

8051 TIMER PROGRAMMING IN ASSEMBLY

By

Dr. Usha Chauhan

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

a. Timer 0 and Timer 1 are 16 bits wide

b. 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

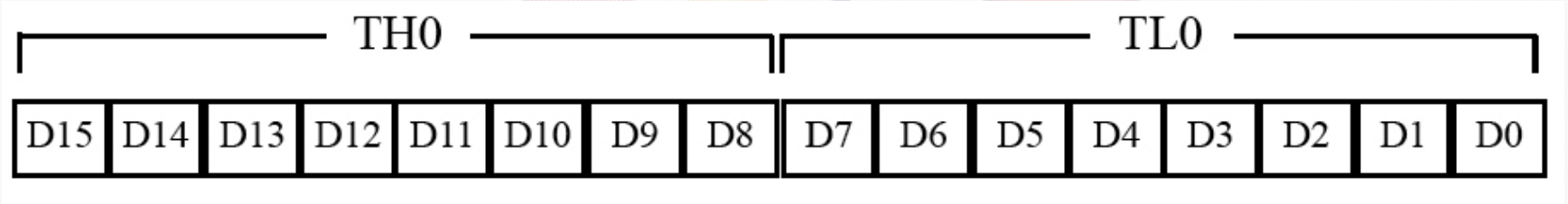

Figure Timer 0 Registers

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

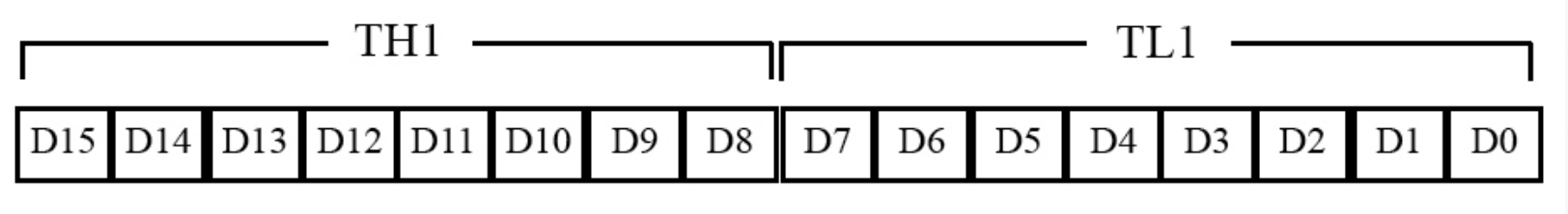

Figure Timer 1 Registers

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 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

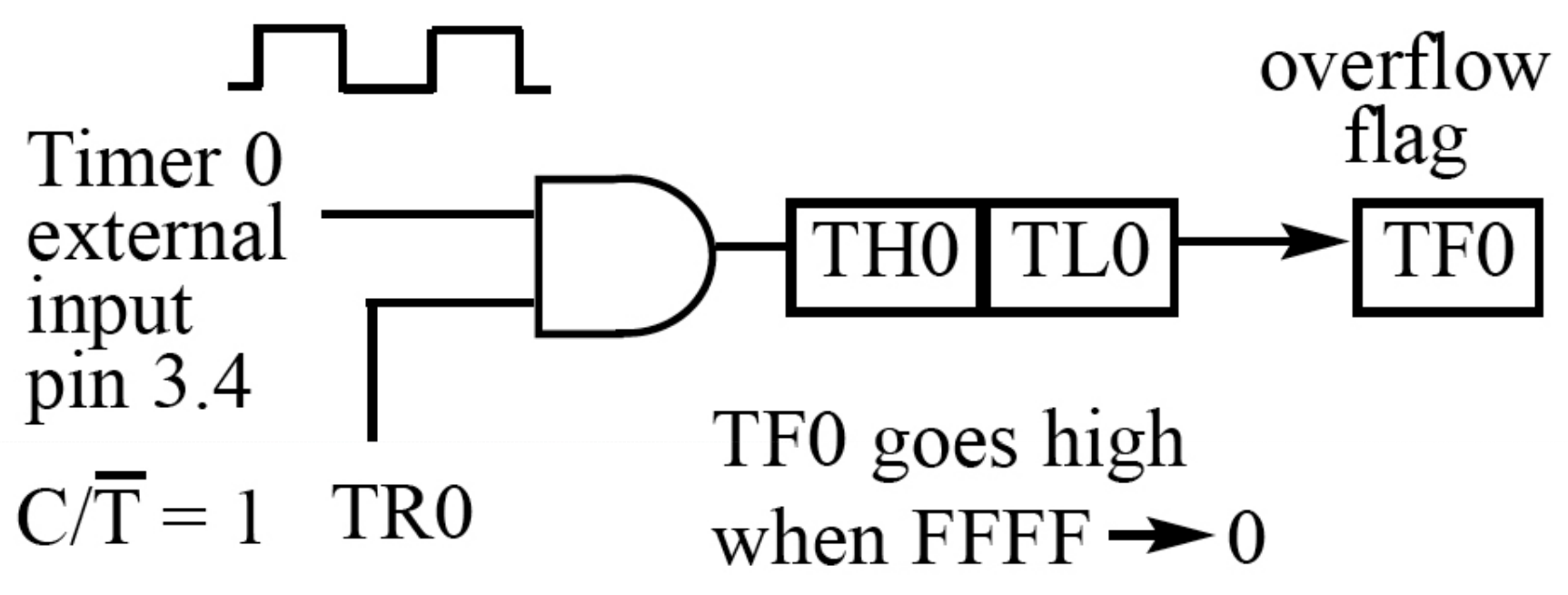

Figure Timer 0 with External Input (Mode 1)

SECTION PROGRAMMING 8051 TIMERS (for information only)

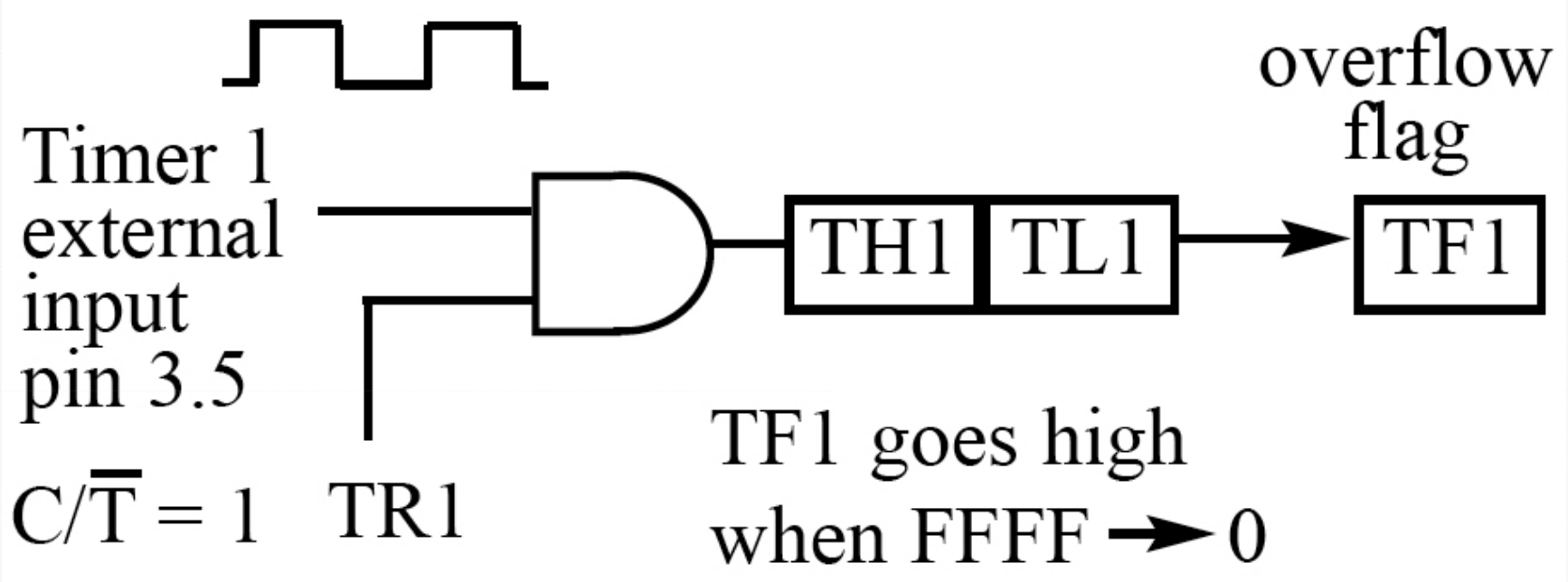

Figure Timer 1 with External Input (Mode 1)

SECTION 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

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

```
01  MOV TMOD,#01            ;Timer 0, mode 1(16-bit mode)
02  HERE: MOV TL0,#0F2H     ;TL0 = F2H, the Low byte
03  MOV TH0,#0FFH           ;TH0 = FFH, the High byte
04  CPL P1.5                ;toggle P1.5
05  ACALL DELAY
06  SJMP HERE               ;load TH, TL again
07
08  DELAY:                  ;delay using Timer 0
09  SETB TR0                ;start Timer 0
10  AGAIN: JNB TF0,AGAIN    ;monitor Timer 0 flag until ;it rolls over
11  CLR TR0                 ;stop Timer 0
12  CLR TF0                 ;clear Timer 0 flag
13  RET
14
15  END
```

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.

```
01  MOV TMOD,#10H                ;Timer 1, mode 1(16-bit)
02  AGAIN: MOV TL1,#34H          ;TL1 = 34H, Low byte
03  MOV TH1,#76H                 ;TH1 = 76H, High byte
04                               ;(7634H = timer value)
05  SETB TR1                     ;start Timer 1
06  BACK: JNB TF1,BACK           ;stay until timer rolls over
07  CLR TR1                      ;stop Timer 1
08  CPL P1.5                     ;comp. P1.5 to get hi, lo
09  CLR TF1                      ;clear Timer 1 flag
10  SJMP AGAIN                   ;reload timer since Mode 1
11                               ;is not auto-reload
12  END
13
14  ;Since FFFFH - 7634H = 89CBH + 1 = 89CCH
15  ;and 89CCH = 35276 clock count.
16  ;35276 x 1.085 us = 38.274 ms for half of the square wave.
17  ;The entire square wave length is 38.274 x 2 = 76.548 ms
18  ;and has a frequency = 13.064 Hz.
19  ;The high and low portions of the
20  ;square wave pulse are equal.
21  ;The overhead due to all the
22  ;instructions in the loop
23  ;is not included.
```