

CHƯƠNG 9

Lập trình cho bộ đếm/ bộ định thời trong 8051

8051 có hai bộ định thời/ bộ đếm. Chúng có thể được dùng như các bộ định thời để tạo một bộ trễ thời gian hoặc như các bộ đếm để đếm các sự kiện xảy ra bên ngoài bộ BVĐK. Trong chương này chúng ta sẽ tìm hiểu về cách lập trình cho chúng và sử dụng chúng như thế nào?

9.1 Lập trình các bộ định thời gian của 8051.

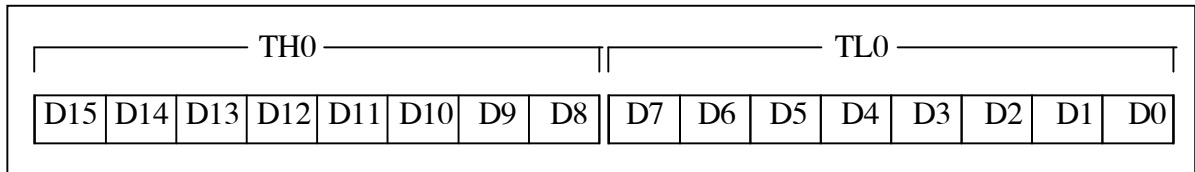
8051 có hai bộ định thời là Timer 0 và Timer1, ở phần này chúng ta bàn về các thanh ghi của chúng và sau đó trình bày cách lập trình chúng như thế nào để tạo ra các độ trễ thời gian.

9.1.1 Các thanh ghi cơ sở của bộ định thời.

Cả hai bộ định thời Timer 0 và Timer 1 đều có độ dài 16 bit được truy cập như hai thanh ghi tách biệt byte thấp và byte cao. Chúng ta sẽ bàn riêng về từng thanh ghi.

9.1.1.1 Các thanh ghi của bộ Timer 0.

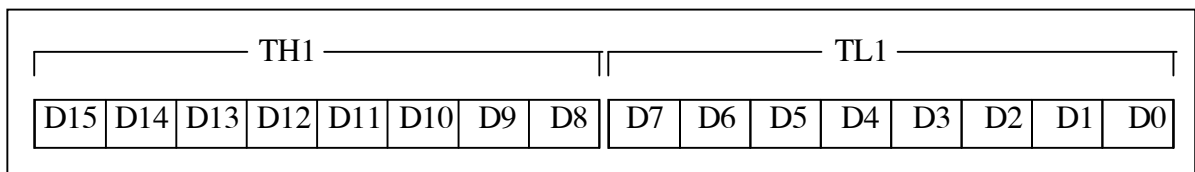
Thanh ghi 16 bit của bộ Timer 0 được truy cập như byte thấp và byte cao. Thanh ghi byte thấp được gọi là TL0 (Timer 0 low byte) và thanh ghi byte cao là TH0 (Timer 0 High byte). Các thanh ghi này có thể được truy cập như mọi thanh ghi khác chẳng hạn như A, B, R0, R1, R2 v.v... Ví dụ, lệnh “MOV TL0, #4FH” là chuyển giá trị 4FH vào TL0, byte thấp của bộ định thời 0. Các thanh ghi này cũng có thể được đọc như các thanh ghi khác. Ví dụ “MOV R5, TH0” là lưu byte cao TH0 của Timer 0 vào R5.



Hình 9.1: Các thanh ghi của bộ Timer 0.

9.1.1.2 Các thanh ghi của bộ Timer 1.

Bộ định thời gian Timer 1 cũng dài 16 bit và thanh ghi 16 bit của nó được chia ra thành hai byte là TL1 và TH1. Các thanh ghi này được truy cập và đọc giống như các thanh ghi của bộ Timer 0 ở trên.



Hình 9.2: Các thanh ghi của bộ Timer 1.

9.1.2 Thanh ghi TMOD (chế độ của bộ định thời).

Cả hai bộ định thời Timer 0 và Timer 1 đều dùng chung một thanh ghi được gọi là IMOD để thiết lập các chế độ làm việc khác nhau của bộ định thời. Thanh ghi TMOD là thanh ghi 8 bit gồm có 4 bit thấp được thiết lập dành cho bộ Timer 0 và 4 bit cao dành cho Timer 1. Trong đó hai bit thấp của chúng dùng để thiết lập chế độ của bộ định thời, còn 2 bit cao dùng để xác định phép toán. Các phép toán này sẽ được bàn dưới đây.

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

Hình 9.3: Thanh ghi IMOD.

9.1.2.1 Các bit M1, M0:

Là các bit chế độ của các bộ Timer 0 và Timer 1. Chúng chọn chế độ của các bộ định thời: 0, 1, 2 và 3. Chế độ 0 là một bộ định thời 13, chế độ 1 là một bộ định thời 16 bit và chế độ 2 là bộ định thời 8 bit. Chúng ta chỉ tập chung vào các chế độ thường được sử dụng rộng rãi nhất là chế độ 1 và 2. Chúng ta sẽ sớm khám phá ra các đặc tính của các chế độ này sau khi khám phần còn lại của thanh ghi TMOD. Các chế độ được thiết lập theo trạng thái của M1 và M0 như sau:

M1	M0	Chế độ	Chế độ hoạt động
0	0	0	Bộ định thời 13 bit gồm 8 bit là bộ định thời/ bộ đếm 5 bit đặt trước
0	1	1	Bộ định thời 16 bit (không có đặt trước)
1	0	2	Bộ định thời 8 bit tự nạp lại
1	1	3	Chế độ bộ định thời chia tách

9.1.2.2 C/ T (đồng hồ/ bộ định thời).

Bit này trong thanh ghi TMOD được dùng để quyết định xem bộ định thời được dùng như một máy tạo độ trễ hay bộ đếm sự kiện. Nếu bit C/T = 0 thì nó được dùng như một bộ định thời tạo độ trễ thời gian. Nguồn đồng hồ cho chế độ trễ thời gian là tần số thạch anh của 8051. Ở phần này chỉ bàn về lựa chọn này, công dụng của bộ định thời như bộ đếm sự kiện thì sẽ được bàn ở phần kế tiếp.

Ví dụ 9.1: Hãy hiển thị xem chế độ nào và bộ định thời nào đối với các trường hợp sau:

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

Lời giải: Chúng ta chuyển đổi giá trị từ số Hex sang nhị phân và đối chiếu hình 93 ta có:

- a) TMOD = 0000 0001, chế độ 1 của bộ định thời Timer 0 được chọn.
b) TMOD = 0010 0000, chế độ 1 của bộ định thời Timer 1 được chọn.
c) TMOD = 0001 0010, chế độ 1 của bộ định thời Timer 0 và chế độ 1 của Timer 1 được chọn.

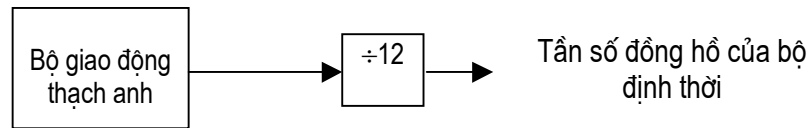
9.1.2.3 Nguồn xung đồng hồ cho bộ định thời:

Như chúng ta biết, mỗi bộ định thời cần một xung đồng hồ để giữ nhịp. Vậy nguồn xung đồng hồ cho các bộ định thời trên 8051 lấy ở đâu? Nếu C/T = 0 thì tần số thạch anh đi liền với 8051 được làm nguồn cho đồng hồ của bộ định thời. Điều đó có nghĩa là độ lớn của tần số thạch anh đi kèm với 8051 quyết định tốc độ nhịp của các bộ định thời trên 8051. Tần số của bộ định thời luôn bằng 1/12 tần số của thạch anh gắn với 8051. Xem ví dụ 9.2.

Ví dụ 9.2:

Hãy tìm tần số đồng bộ và chu kỳ của bộ định thời cho các hệ dựa trên 8051 với các tần số thạch anh sau:

- a) 12MHz
- b) 16MHz
- c) 11,0592MHz



Lời giải:

- a) $\frac{1}{12} \times 12\text{MHz} = 1\text{MHz}$ và $T = \frac{1}{1/1\text{MHz}} = 1\mu\text{s}$
- b) $\frac{1}{12} \times 16\text{MHz} = 1,333\text{MHz}$ và $T = \frac{1}{1,333\text{MHz}} = 0,75\mu\text{s}$
- c) $\frac{1}{12} \times 11,0592\text{MHz} = 921,6\text{kHz}$ và $T = \frac{1}{0,9216\text{MHz}} = 1,085\mu\text{s}$

Mặc dù các hệ thống dựa trên 8051 khác với tần số thạch anh từ 10 đến 40MHz, song ta chỉ tập chung vào tần số thạch anh 11,0592MHz. Lý do đằng sau một số lẻ như vậy là hải làm việc với tần suất boudit đối với truyền thông nối tiếp của 8051. Tần số XTAL = 11,0592MHz cho phép hệ 8051 truyền thông với IBM PC mà không có lỗi, điều mà ta sẽ biết ở chương 10.

9.1.3 Bít cổng GATE.

Một bít khác của thanh ghi TMOD là bít cổng GATE. Để ý trên hình 9.3 ta thấy cả hai bộ định thời Timer0 và Timer1 đều có bít GATE. Vậy bít GATE dùng để làm gì? Mỗi bộ định thời thực hiện điểm khởi động và dừng. Một số bộ định thời thực hiện điều này bằng phần mềm, một số khác bằng phần cứng và một số khác vừa bằng phần cứng vừa bằng phần mềm. Các bộ định thời trên 8051 có cả hai. Việc khởi động và dừng bộ định thời được khởi động bằng phần mềm bởi các bít khởi động bộ định thời TR là TR0 và TR1. Điều này có được nhờ các lệnh “SETB TR1” và “CLR TR1” đối với bộ Timer1 và “SETB TR0” và “CLR TR0” đối với bộ Timer0. Lệnh SETB khởi động bộ định thời và lệnh CLR dùng để dừng nó. Các lệnh này khởi động và dừng các bộ định thời khi bít GATE = 0 trong thanh ghi TMOD. Khởi động và ngừng bộ định thời bằng phần cứng từ nguồn ngoài bằng cách đặt bít GATE = 1 trong thanh ghi TMOD. Tuy nhiên, để tránh sự lẫn lộn ngay từ bây giờ ta đặt GATE = 0 có nghĩa là không cần khởi động và dừng các bộ định thời bằng phần cứng từ bên ngoài. Để sử dụng phần mềm để khởi động và dừng các bộ định thời phần mềm để khởi động và dừng các bộ định thời khi GATE = 0. Chúng ta chỉ cần các lệnh “SETB TRx” và “CLR TRx”. Việc sử dụng phần cứng ngoài để khởi động và dừng bộ định thời ta sẽ bàn ở chương 11 khi bàn về các ngắt.

Ví dụ 9.3:

Tìm giá trị cho TMOD nếu ta muốn lập trình bộ Timer0 ở chế độ 2 sử dụng thạch anh XTAL 8051 làm nguồn đồng hồ và sử dụng các lệnh để khởi động và dừng bộ định thời.

Lời giải:

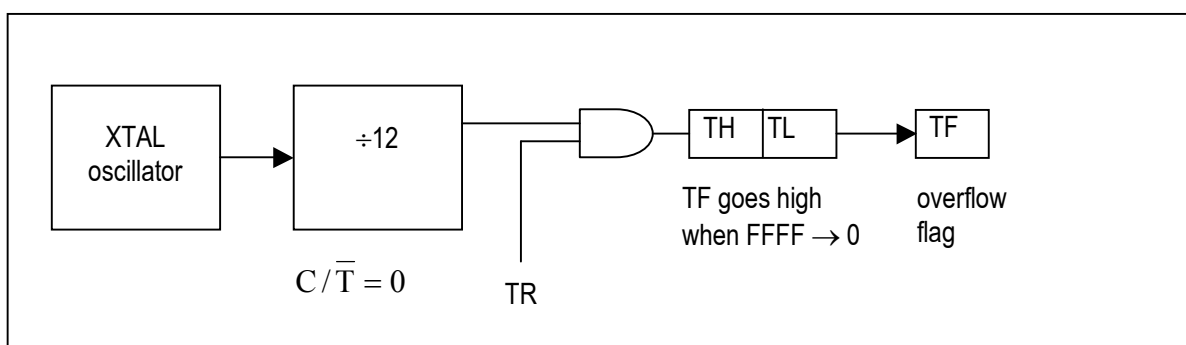
TMOD = 0000 0010: Bộ định thời Timer0, chế độ 2 C/T = 0 dùng nguồn XTAL GATE = 0 để dùng phần mềm trong để khởi động và dừng bộ định thời.

Như vậy, bây giờ chúng ta đã có hiểu biết cơ bản về vai trò của thanh ghi TMOD, chúng ta sẽ xét chế độ của bộ định thời và cách chúng được lập trình như thế nào để tạo ra một độ trễ thời gian. Do chế độ 1 và chế độ 2 được sử dụng rộng rãi nên ta đi xét chi tiết từng chế độ một.

9.1.4 Lập trình cho mỗi chế độ Mode1.

Dưới đây là những đặc tính và những phép toán của chế độ Mode1:

1. Nó là bộ định thời 16 bit, do vậy nó cho phép các giá trị 0000 đến FFFFH được nạp vào các thanh ghi TL và TH của bộ định thời.
2. Sau khi TL và TH được nạp một giá trị khởi tạo 16 bit thì bộ định thời phải được khởi động. Điều này được thực hiện bởi “SETB TR0” đối với Timer 0 và “SETB TR1” đối với Timer1.
3. Sau khi bộ định thời được khởi động, nó bắt đầu đếm lên. Nó đếm lên cho đến khi đạt được giới hạn FFFFH của nó. Khi nó quay qua từ FFFFH về 0000 thì nó bật lên bit cờ TF được gọi là cờ bộ định thời. Cờ bộ định thời này có thể được hiển thị. Khi cờ bộ định thời này được thiết lập từ một trong các phương án để dừng bộ định thời bằng các lệnh “CLR TR0” đối với Timer0 hoặc “CLR TR1” đối với Timer1. ở đây cũng cần phải nhắc lại là đối với bộ định thời đều có cờ TF riêng của mình: TF0 đối với Timer0 và TF1 đối với Timer1.



4. Sau khi bộ định thời đạt được giới hạn của nó và quay quan giá trị FFFFH, muốn lặp lại quá trình thì các thanh ghi TH và TL phải được nạp lại với giá trị ban đầu và TF phải được duy trì về 0.

9.1.4.1 Các bước lập trình ở chế độ Mode 1.

Để tạo ra một độ trễ thời gian dùng chế độ 1 của bộ định thời thì cần phải thực hiện các bước dưới đây.

1. Nạp giá trị TMOD cho thanh ghi báo độ định thời nào (Timer0 hay Timer1) được sử dụng và chế độ nào được chọn.
2. Nạp các thanh ghi TL và TH với các giá trị đếm ban đầu.
3. Khởi động bộ định thời.
4. Duy trì hiển thị cờ bộ định thời TF bằng lệnh “JNB TFx, đích” để xem nó được bật không. Thoát vòng lặp khi TF được lên cao.
5. Dừng bộ định thời.
6. Xoá cờ TF cho vòng kế tiếp.
7. Quay trở lại bước 2 để nạp lại TL và TH.

Để tính toán thời gian trễ chính xác và tần số sóng vuông được tạo ra trên chân P1.5 thì ta cần biết tần số XTAL (xem ví dụ 9.5).

Từ ví dụ 9.6 ta có thể phát triển một công thức tính toán độ trễ sử dụng chế độ Mode1 (16 bit) của bộ định thời đối với tần số thạch anh XTAL = 11, 0592MHz (xem hình 9.4). Máy tính trong thư mục Accessrry của Microsoft Windows có thể giúp ta tìm các giá trị TH và TL. Máy tính này hỗ trợ các phép tính theo số thập phân, nhị phân và thập lục.

a) Tính theo số Hex	b) Tính theo số thập phân
(FFFF - YYXX + 1). 1,085 μ s trong đó YYXX là các giá trị khởi tạo của TH, TL tương ứng. Lưu ý rằng các giá trị YYXX là theo số Hex.	Chuyển đổi các giá trị YYXX của TH, TL về số thập phân để nhận một số thập phân NNNNN sau đó lấy (65536 - NNNNN).1,085 μ s.

Hình 9.4: Công thức tính toán độ trễ thời gian đối với tần số XTAL = 11, 0592MHz.

Ví dụ 9.4:

Trong chương trình dưới đây ta tạo ra một sóng vuông với độ đầy xung 50% (cùng tỷ lệ giữa phần cao và phần thấp) trên chân P1.5. Bộ định thời Timer0 được dùng để tạo độ trễ thời gian. Hãy phân tích chương trình này.

```

HERE:      MOV    TMOD, #01          ; Sử dụng Timer0 và chế độ 1(16 bit)
            MOV    TL0, #0F2H        ; TL0 = F2H, byte thấp
            MOV    TH0, #0FFH        ; TH0 = FFH, byte cao
            CPL     P1.5              ; Sử dụng chân P1.5
            ACALL  DELAY
            SJMP   HERE              ; Nạp lại TH, TL
; _____ delay using timer0.
DELAY:
            SETB   TR0                ; Khởi động bộ định thời Timer0
AGAIN:      JNB    TF0, AGAIN         ; Hiển thị cờ bộ định thời cho đến khi nó vượt qua FFFFH.
            CLR     TR0               ; Dừng bộ Timer
            CLR     TF0               ; Xóa cờ bộ định thời 0
            RET

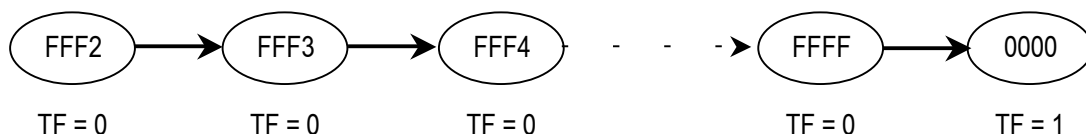
```

Lời giải:

Trong chương trình trên đây chú ý các bước sau:

1. TMOD được nạp.
2. Giá trị FFF2H được nạp và TH0 - TL0
3. Chân P1.5 được chọn dùng cho phần cao thấp của xung.
4. Chương trình con DELAY dùng bộ định thời được gọi.
5. Trong chương trình con DELAY bộ định thời Timer0 được khởi động bởi lệnh “SETB TR0”
6. Bộ Timer0 đếm lên với mỗi xung đồng hồ được cấp bởi máy phát thạch anh. Khi bộ định thời đếm tăng qua các trạng thái FFF3, FFF4 ... cho đến khi đạt giá trị FFFFH. Và một xung nữa là nó quay về không và bật cờ bộ định thời TF0 = 1. Tại thời điểm này thì lệnh JNB hạn xuống.
7. Bộ Timer0 được dừng bởi lệnh “CLR TR0”. Chương trình con DELAY kết thúc và quá trình được lặp lại.

Lưu ý rằng để lặp lại quá trình trên ta phải nạp lại các thanh ghi TH và TL và khởi động lại bộ định thời với giả thiết tần số XTAL = 11, 0592MHz.



Ví dụ 9.5:

Trong ví dụ 9.4 hãy tính toán lượng thời gian trễ trong chương trình con DELAY được tạo ra bởi bộ định thời với giá thiết tần số XTAL = 11,0592MHz.

Lời giải:

Bộ định thời làm việc với tần số đồng hồ bằng 1/12 tần số XTAL, do vậy ta có $\frac{11,0592}{12} = 0,9216\text{MHz}$ là tần số của bộ định thời. Kết quả là mỗi nhịp xung đồng hồ có

chu kỳ $T = \frac{1}{0,9216\text{MHz}} = 1,085\mu\text{s}$. Hay nói cách khác, bộ Timer0 đếm tăng sau 1,085μs

để tạo ra bộ trễ bằng số đếm $\times 1,085\mu\text{s}$.

Số đếm bằng FFFFH - FFF2H = 0DH (13 theo số thập phân). Tuy nhiên, ta phải cộng 1 vào 13 vì cần thêm một nhịp đồng hồ để nó quay từ FFFFH về 0 và bật cờ TF. Do vậy, ta có $14 \times 1,085\mu\text{s} = 15,19\mu\text{s}$ cho nửa chu kỳ và cả chu kỳ là $T = 2 \times 15,19\mu\text{s} = 30,38\mu\text{s}$ là thời gian trễ được tạo ra bởi bộ định thời.

Ví dụ 9.6:

Trong ví dụ 9.5 hãy tính toán tần số của xung vuông được tạo ra trên chân P1.5.

Lời giải:

Trong tính toán độ thời gian trễ của ví dụ 9.5 ta không tính đến tổng phí của các lệnh trong vòng lặp. Để tính toán chính xác hơn ta cần bổ xung thêm các chu kỳ thời gian của các lệnh trong vòng lặp. Để làm điều đó ta sử dụng các chu kỳ máy từ bảng A-1 trong phụ lục Appendix A được chỉ dưới đây.

HERE:	MOV	TL0, #0F2H	2
	MOV	TH0, #0FFH	2
	CPL	P1.5	1
	ACALL	DELAY	2
	SJMP	HERE	2
; _____delay using timer0			
DELAY:	SETB	TR0	1
AGAIN:	JNB	TF0, AGAIN	1
	CLR	TR0	1
	CLR	TF0	1
	RET		1
Total			27

$T = (2 \times 27 \times 1.085\mu\text{s})$ and $F = 17067.75\text{Hz}$.

Tổng số chu kỳ đã bổ xung là x7 nên chu kỳ thời gian trễ là $T = 2 \times 27 \times 1.085\mu\text{s} = 58,59\mu\text{s}$ và tần số là $F = 17067,75\text{Hz}$.

Ví dụ 9.7:

Hãy tìm ra độ trễ được tạo ra bởi Timer0 trong đoạn mã sau sử dụng cả hai phương pháp của hình 9.4. Không tính các tổng phí của các lệnh.

	CLR	P2.3	; Xóa P2.3
	MOV	TMOD, #01	; Chọn Timer0, chế độ 1 (16 bit)
HERE:	MOV	TL0, #3EH	; TL0 = 3EH, byte thấp
	MOV	TH0, #0B8H	; TH0 = B8H, byte cao
	SETB	P2.3	; Bật P2.3 lên cao
	SETB	TR0	; Khởi động Timer0
AGAIN:	JNB	TF0, AGAIN	; Hiển thị cờ bộ định thời TF0

CLR	TR0	; Dừng bộ định thời.
CLR	TF0	; Xoá cờ bộ định thời cho vòng sau
CLR	P2.3	

Lời giải:

a) Độ trễ được tạo ra trong mã trên là:

$(FFFF - B83E + 1) = 47C2H = 18370$ hệ thập phân $18370 \times 1,085\mu s = 19,93145\mu s$.

b) Vì $TH - TL = B83EH = 47166$ (số thập phân) ta có $65536 - 47166 = 18370$.

Điều này có nghĩa là bộ định thời gian đếm từ B83EH đến FFFF. Nó được cộng với một số đếm để về 0 thành một bộ tổng là $18370\mu s$. Do vậy ta có $18370 \times 1,085\mu s = 19,93145ms$ là độ rộng xung.

Ví dụ 9.8:

Sửa giá trị của TH và TL trong ví dụ 9.7 để nhận được độ trễ thời gian lớn nhất có thể. Hãy tính độ trễ theo miligiây. Trong tính toán cần đưa vào cả tổng phí của các lệnh.

Để nhận độ trễ thời gian lớn nhất có thể ta đặt TH và TL bằng 0. Điều này làm cho bộ định thời đếm từ 0000 đến FFFFH và sau đó quay qua về 0.

	CLR	P2.3	; Xoá P2.3
	MOV	TMOD, #01	; Chọn Timer0, chế độ 1 (16 bit)
HERE:	MOV	TL0, #0	; Đặt TL0 = 0, byte thấp
	MOV	TH0, #0	; Đặt TH0 = 0, byte cao
	SETB	P2.3	; Bật P2.3 lên cao
	SETB	TR0	; Khởi động bộ Timer0
AGAIN:	JNB	TF0, AGAIN	; Hiển thị cờ bộ định thời TF0
	CLR	TR0	; Dừng bộ định thời.
	CLR	TF0	; Xoá cờ TF0
	CLR	P2.3	

Thực hiện biến TH và TL bằng 0 nghĩa là bộ định thời đếm tăng từ 0000 đến FFFFH và sau đó quay qua về 0 để bật cờ bộ định thời TF. Kết quả là nó đi qua 65536 trạng thái. Do vậy, ta có độ trễ $= (65536 - 0) \times 1.085\mu s = 71.1065\mu s$.

Trong ví dụ 9.7 và 9.8 chúng ta đã không nạp lại TH và TL vì nó là một xung đơn. Xét ví dụ 9.9 dưới đây để xem việc nạp lại làm việc như thế nào ở chế độ 1.

Ví dụ 9.9:

Chương trình dưới đây tạo ra một sóng vuông trên chân P2.5 liên tục bằng việc sử dụng bộ Timer1 để tạo ra độ trễ thời gian. Hãy tìm tần số của sóng vuông nếu tần số XTAL = 11.0592MHz. Trong tính toán không đưa vào tổng phí của các lệnh vòng lặp:

	MOV	TMOD, #01H	; Chọn Timer0, chế độ 1 (16 bit)
HERE:	MOV	TL1, #34H	; Đặt byte thấp TL1 = 34H
	MOV	TH0, #76H	; Đặt byte cao TH1 = 76H
			; (giá trị bộ định thời là 7634H)
	SETB	TR1	; Khởi động bộ Timer1
AGAIN:	JNB	TF1, BACK	; Ở lại cho đến khi bộ định thời đếm qua 0
	CLR	TR1	; Dừng bộ định thời.
	CPL	P1.5	; Bù chân P1.5 để nhận Hi, L0
	CLR	TF	; Xoá cờ bộ định thời
	SJMP	AGAIN	; Nạp lại bộ định thời do chế độ 1 không tự động nạp lại.

Lời giải:

Trong chương trình trên đây ta lưu ý đến đích của SJMP. Ở chế độ 1 chương trình phải nạp lại thanh ghi. TH và TL mỗi lần nếu ta muốn có sóng dạng liên tục. Dưới đây là kết quả tính toán:

Vì $FFFFH - 7634H = 89CBH + 1 = 89CCH$ và $90CCH = 35276$ là số lần đếm xung đồng hồ, độ trễ là $35276 \times 1.085\mu s = 38274ms$ và tần số là $\frac{1}{38274}(Hz) = 26127Hz$.

Cũng để ý rằng phần cao và phần thấp của xung sóng vuông là bằng nhau. Trong tính toán trên đây là chưa kể đến tổng phí các lệnh vòng lặp.

9.1.4.2 Tìm các giá trị cần được nạp vào bộ định thời.

giả sử rằng chúng ta biết lượng thời gian trễ mà ta cần thì câu hỏi đặt ra là làm thế nào để tìm ra được các giá trị cần thiết cho các thanh ghi TH và TL. Để tính toán các giá trị cần được nạp vào các thanh ghi TH và TL chúng ta hãy nhìn vào ví dụ sau với việc sử dụng tần số dao động XTAL = 11.0592MHz đối với hệ 8051.

Từ ví dụ 9.10 ta có thể sử dụng những bước sau để tìm các giá trị của các thanh ghi TH và TL.

1. Chia thời gian trễ cần thiết cho $1.0592\mu s$
2. Thực hiện $65536 - n$ với n là giá trị thập phân nhận được từ bước 1.
3. Chuyển đổi kết quả ở bước 2 sang số Hex với yxxx là giá trị .hex ban đầu cần phải nạp vào các thanh ghi bộ định thời.
4. Đặt TL = xx và TH = yy.

Ví dụ 9.10:

Giả sử tần số XTAL = 11.0592MHz. Hãy tìm các giá trị cần được nạp vào các thanh ghi vào các thanh ghi TH và TL nếu ta muốn độ thời gian trễ là $5\mu s$. Hãy trình bày chương trình cho bộ Timer0 để tạo ra bộ xung với độ rộng $5\mu s$ trên chân P2.3.

Lời giải:

Vì tần số XTAL = 11.0592MHz nên bộ đếm tăng sau mỗi chu kỳ $1.085\mu s$. Điều đó có nghĩa là phải mất rất nhiều khoảng thời gian $1,085\mu s$ để có được một xung $5\mu s$. Để có được ta chia $5ms$ cho $1.085\mu s$ và nhận được số $n = 4608$ nhịp. Để nhận được giá trị cần được nạp vào TL và TH thì ta tiến hành lấy 65536 trừ đi 4608 bằng 60928 . Ta đổi số này ra số hex thành $EE00H$. Do vậy, giá trị nạp vào TH là EE Và TL là 00 .

	CLR	P2.3	; Xóa bit P2.3
	MOV	TMOD, #01	; Chọn Timer0, chế độ 1 (16 bit)
HERE:	MOV	TL0, #0	; Nạp TL = 00
	MOV	TH0, #EEH	; Nạp TH = EEH
	SETB	P2.3	; Bật P2.3 lên cao
	SETB	TR0	; Khởi động bộ định thời Timer0
AGAIN:	JNB	TF0, AGAIN	; Hiển thị cờ TF0 cho đến khi bộ đếm quay về 0
	CLR	TR0	; Dừng bộ định thời.
	CLR	TF0	; Xóa cờ TF0 cho vòng sau.

Ví dụ 9.11:

Giả sử ta có tần số XTAL là 11,0592MHz hãy viết chương trình tạo ra một sóng vuông tần số 2kHz trên chân P2.5.

Đây là trường hợp giống với ví dụ 9.10 ngoài trừ một việc là ta phải chọn bit để tạo ra sóng vuông. Xét các bước sau:

- a) $T = \frac{1}{f} = \frac{1}{2\text{kHz}} = 500\mu\text{s}$ là chu kỳ của sóng vuông.
- b) Khoảng thời gian cao và phần thấp là $\frac{1}{2}T$ bằng $250\mu\text{s}$.
- c) Số nhịp cần trong thời gian đó là $\frac{250\mu\text{s}}{1,085\mu\text{s}} = 230$ và giá trị cần nạp vào các thanh ghi cần tìm là $65536 - 230 = 65306$ và ở dạng hex là FF1AH.
- d) giá trị nạp vào TL là 1AH và TH là FFH.
Chương trình cần viết là:

```

AGAIN:    MOV    TMOD, #10H        ; Chọn bộ định thời Timer0, chế độ 1 (16 bit)
          MOV    TL1, #1AH         ; Gán giá trị byte thấp TL1 = 1AH
          MOV    TH1, #0FFH        ; Gán giá trị byte cao TH1 = FFH
          SETB   TR1               ; Khởi động Timer1
BACK:     JNB    TF1, BACK          ; giữ nguyên cho đến khi bộ định thời quay về 0
          CLR    TR1               ; Dừng bộ định thời.
          CPL    P1.5              ; Bù bit P1.5 để nhận giá trị cao, thấp.
          CLR    TF1               ; Xoá cờ TF1
          SUMP  AGAIN              ; Nạp lại bộ định thời vì chế độ 1 không tự nạp
                                   lại.

```

Ví dụ 9.12:

Trước hết ta thực hiện các bước sau:

- a) Tính chu kỳ sóng vuông: $T = \frac{1}{50\text{Hz}} = 20\mu\text{s}$
- b) Tính thời gian nửa chu kỳ cho phần cao: $\frac{1}{2}T = 10\mu\text{s}$
- c) Tính số nhịp đồng hồ: $n = \frac{10\mu\text{s}}{1,085\mu\text{s}} = 9216$
- d) Tính giá trị cần nạp vào TH và TL: $65536 - 9216 = 56320$ chuyển về dạng Hex là DC00H và TH = DCH và TL = 00H.

```

AGAIN:    MOV    TMOD, #10H        ; Chọn bộ định thời Timer0, chế độ 1 (16 bit)
          MOV    TL1, #00          ; Gán giá trị byte thấp TL1 = 00
          MOV    TH1, #0DCH        ; Gán giá trị byte cao TH1 = DC
          SETB   TR1               ; Khởi động Timer1
BACK:     JNB    TF1, BACK          ; giữ nguyên cho đến khi bộ định thời quay về 0
          CLR    TR1               ; Dừng bộ định thời.
          CPL    P2.3              ; Bù bit P1.5 để nhận giá trị cao, thấp.
          CLR    TF1               ; Xoá cờ TF1
          SUMP  AGAIN              ; Nạp lại bộ định thời vì chế độ 1 không tự nạp
                                   lại.

```

9.1.4.3 Tạo một độ trễ thời gian lớn.

Như ta đã biết từ các ví dụ trên là lượng thời gian trễ cần tạo ra phụ thuộc vào hai yếu tố:

- Tần số thạch anh XTAL
- Thanh ghi 16 bit của bộ định thời ở chế độ 1

Cả hai yếu tố này nằm ngoài khả năng điều chỉnh của lập trình viên 8051. Ví như ta đã biết giá trị lớn nhất của độ trễ thời gian có thể đạt được bằng cách đặt cả TH và TL

bằng 0. Nhưng điều này xảy ra khi như vậy đều không đủ? Ví dụ 9.13 dưới đây cách làm thế nào để có giá trị độ trễ thời gian lớn.

9.1.4.4 Sử dụng bàn tính của Windows để tìm TH và TL.

Bàn tính Calculator của Windows có ngay trong máy tính PC của chúng ta và rất dễ sử dụng để tìm ra các giá trị cho TH và TL. Giả sử tìm giá trị cho TH và TL với độ trễ thời gian lớn là 35.000 nhịp đồng hồ với chu kỳ 1,085 μ s. Ta thực hiện các bước như sau:

1. Chọn máy tính Calculator từ Windows và đặt chế độ tính về số thập phân Decimal.
2. Nhập số 35.000 vào từ bàn phím.
3. Chuyển về chế độ Hex trên Calculator nó cho ta giá trị 88B8H.
4. Chọn +/- để nhận số đổi dấu - 35.000 dạng thập phân và chuyển về dạng Hex là 7748H.
5. Hai số hex cuối là cho TL = 48 và hai số Hex tiếp theo là cho TH = 77. Ta bỏ quan các số F ở phía bên phải trên Calculator vì số của ta là 16 bit.

Ví dụ 9.13:

Hãy kiểm tra chương trình sau và tìm độ trễ thời gian theo giây, không tính đến tổng phí các lệnh trong vòng lặp.

	MOV	TMOD, #10H	; Chọn bộ Timer1, chế độ 1 (16 bit)
AGAIN:	MOV	R3, #200	; Chọn bộ đếm độ giữ chậm lớn
	MOV	TL1, #08	; Nạp byte thấp TL1 = 08
	MOV	TH1, #08	; Nạp byte cao TH1 = 01
	SETB	TR1	; Khởi động Timer1
BACK:	JNB	TF1, BACK	; giữ nguyên cho đến khi bộ định thời quay về 0
	CLR	TR1	; Dừng bộ định thời.
	CLR	TF1	; Xoá cờ bộ định thời TF1
	DJNZ	R3, AGAIN	; Nếu R3 không bằng không thì nạp lại bộ định thời.

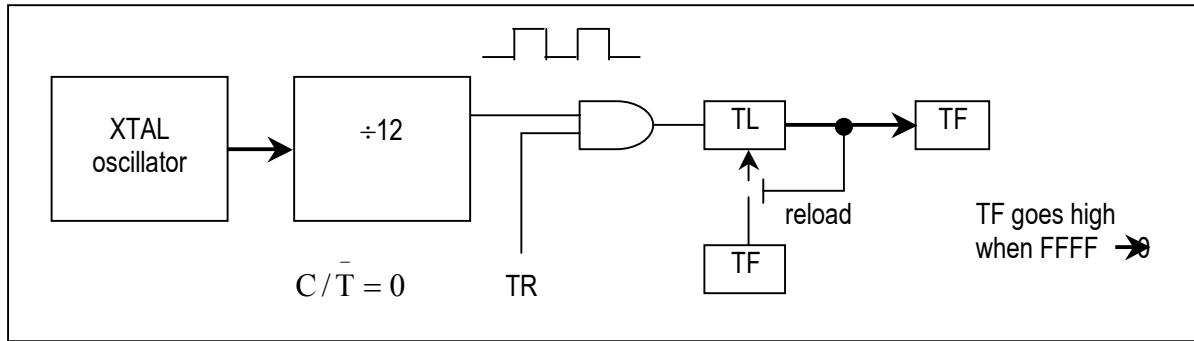
9.1.5 Chế độ 0.

Chế độ 0 hoàn toàn giống chế độ 1 chỉ khác là bộ định thời 16 bit được thay bằng 13 bit. Bộ đếm 13 bit có thể giữ các giá trị giữa 0000 đến 1FFFF trong TH - TL. Do vậy khi bộ định thời đạt được giá trị cực đại của nó là 1FFFFH thì nó sẽ quay trở về 0000 và cờ TF được bật lên.

9.1.6 Lập trình chế độ 2.

Các đặc trưng và các phép tính của chế độ 2:

1. Nó là một bộ định thời 8 bit, do vậy nó chỉ cho phép các giá trị từ 00 đến FFH được nạp vào thanh ghi TH của bộ định thời.
2. Sau khi TH được nạp với giá trị 8 bit thì 8051 lấy một bản sao của nó đưa vào TL. Sau đó bộ định thời phải được khởi động. Điều này được thực hiện bởi lệnh "SETB TR0" đối với Timer0 và "SETB TR1" đối với Timer1 giống như ở chế độ 1.
3. Sau khi bộ định thời được khởi động, nó bắt đầu đếm tăng lên bằng cách tăng thanh ghi TL. Nó đếm cho đến khi đạt giá trị giới hạn FFH của nó. Khi nó quay trở về 00 từ FFH, nó thiết lập cờ bộ định thời TF. Nếu ta sử dụng bộ định thời Timer0 thì đó là cờ TF0, còn Timer1 thì đó là cờ TF1.



4. Khi thanh ghi TL quay trở về 00 từ FFH thì TF được bật lên 1 thì thanh ghi TL được tự động nạp lại với giá trị ban đầu được giữ bởi thanh ghi TH. Để lặp lại quá trình chúng ta đơn giản chỉ việc xoá cờ TF và để cho nó chạy mà không cần sự can thiệp của lập trình viên để nạp lại giá trị ban đầu. Điều này làm cho chế độ 2 được gọi là chế độ từ nạp lại so với chế độ 1 thì ta phải nạp lại các thanh ghi TH và TL.

Cần phải nhấn mạnh rằng, chế độ 2 là bộ định thời 8 bit. Tuy nhiên, nó lại có khả năng tự nạp khi tự nạp lại thì TH thực chất là không thay đổi với giá trị ban đầu được giữ nguyên, còn TL được nạp lại giá trị được sao từ TH. Chế độ này có nhiều ứng dụng bao gồm việc thiết lập tần số baud trong truyền thông nối tiếp như ta sẽ biết ở chương 10.

9.1.5.1 Các bước lập trình cho chế độ 2.

Để tạo ra một thời gian trễ sử dụng chế độ 2 của bộ định thời cần thực hiện các bước sau:

1. Nạp thanh ghi giá trị TMOD để báo bộ định thời gian nào (Timer0 hay Timer1) được sử dụng và chế độ làm việc nào của chúng được chọn.
2. Nạp lại các thanh ghi TH với giá trị đếm ban đầu.
3. Khởi động bộ định thời.
4. Duy trì hiển thị cờ bộ định thời TF sử dụng lệnh “JNB TFx, đích” để xem nó sẽ được bật chưa. Thoát vòng lặp khi TF lên cao.
5. Xoá cờ TF.
6. Quay trở lại bước 4 vì chế độ 2 là chế độ tự nạp lại.

Ví dụ 9.14 minh hoạ những điều này. Để có được độ trễ lớn chúng ta có thể dùng nhiều thanh ghi như được chỉ ra trong ví dụ 9.15.

Ví dụ 9.14:

Giả sử tần số XTAL = 11.0592MHz. Hãy tìm a) tần số của sóng vuông được tạo ra trên chân P1.0 trong chương trình sau và b) tần số nhỏ nhất có thể có được bằng chương trình này và giá trị TH để đạt được điều đó.

MOV	TMOD, #20H	; Chọn Timer1/ chế độ 2/ 8 bit/ tự nạp lại.
MOV	TH1, #5	; TH1 = 5
SETB	TR1	; Khởi động Timer1
BACK:	JNB TF1, BACK	; giữ nguyên cho đến khi bộ định thời quay về 0
	CPL P1.0	; Dừng bộ định thời.
	CLR TF1	; Xoá cờ bộ định thời TF1
	SJMP BACK	; Chế độ 2 tự động nạp lại.

Lời giải:

- a) Trước hết để ý đến đích của lệnh SJMP. Trong chế độ 2 ta không cần phải nạp lại TH vì nó là chế độ tự nạp. Bây giờ ta lấy $(256 - 05) \cdot 1.085\mu s = 251 \cdot 1.085\mu s = 272.33\mu s$ là phần cao của xung. Cả chu kỳ của xung là $T = 544.66\mu s$ và tần số là $\frac{1}{T} = 1,83597\text{kHz}$.
- b) Để nhận tần số nhỏ nhất có thể ta cần tạo T chu kỳ lớn nhất có thể có nghĩa là TH = 00. Trong trường hợp này ta có $T = 2 \times 256 \times 1.085\mu s = 555.52\mu s$ và tần số nhỏ nhất sẽ là $\frac{1}{T} = 1,8\text{kHz}$.

Ví dụ 9.15:

Hãy tìm tần số của xung vuông được tạo ra trên P1.0.

Lời giải:

```

AGAIN:    MOV    TMOD, #2H           ; Chọn Timer0, chế độ 1 (8 bit tự nạp lại)
          MOV    TH0, #0             ; Nạp TH0 = 00
          MOV    R5, #250            ; Đếm cho độ trễ lớn
          ACALL DELAY
          CPL     P1.0
          SJMP   AGAIN
DELAY:    SETB   TR0                 ; Khởi động Timer0
BACK:     JNB    TF1, BACK            ; giữ nguyên cho đến khi bộ định thời quay về 0
          CLR    TR0                 ; Dừng Timer0.
          CLR    TF0                 ; Xoá cờ TF0 cho vòng sau.
          DJNZ   R5, DELAY
          RET

```

$$T = 2 \times (250 \times 256 \times 1.085\mu s) = 1.38.88\text{ms và } f = 72\text{Hz}.$$

Ví dụ 9.16:

Giả sử ta đang lập trình chế độ 2 hãy tìm các giá trị (dạng Hex) cần nạp vào TH cho các trường hợp sau:

- a) MOV TH1, #200 b) MOV TH0, #60
 c) MOV TH1, #3 d) MOV TH1, #12
 e) MOV TH0, #48

Lời giải:

Chúng ta có thể sử dụng bàn tính Calculator của Windows để kiểm tra kết quả được cho bởi trình hợp ngữ. Hãy chọn Calculator ở chế độ Decimal và nhập vào số 200. Sau đó chọn Hex, rồi ấn +/- để nhận giá trị của TH. Hãy nhớ rằng chúng ta chỉ sử dụng đúng hai chữ số và bỏ qua phần bên trái vì dữ liệu chúng ta là 8 bit. Kết quả ta nhận được như sau:

Dạng thập phân

- 200
 - 60
 - 3
 - 12
 - 48

Số bù hai (giá trị TH)

38H
 C4H
 FDH
 F4H
 DOH

9.1.5.2 Các trình hợp ngữ và các giá trị âm.

Vì bộ định thời là 8 bit trong chế độ 2 nên ta có thể để cho trình hợp ngữ tính giá trị cho TH. Ví dụ, trong lệnh “MOV TH0, # - 100” thì trình hợp ngữ sẽ tính toán $-100 = 9C$ và gán TH = 9CH. Điều này làm cho công việc của chúng ta dễ dàng hơn.

Ví dụ 9.17:

Hãy tìm a) tần số sóng vuông được tạo ra trong đoạn mã dưới đây và độ đầy xung của sóng này.

```

MOV    TMOD, #2H           ; Chọn bộ Timer0/ chế độ 2/ (8 bit, tự nạp lại).
MOV    TH0, # - 150        ; Nạp TH0 = 6AH là số bù hai của - 150
SETB   TR1                 ; Khởi động Timer1
AGAIN: SETB P1.3            ; P1.3 = 1
        ACALL DELAY
        ACALL P1.3          ; P1.3 = 0
        ACALL DELAY
        SJMP AGAIN

        SETB   TR0          ; Khởi động Timer0
BACK:   JNB    TF0, BACK    ; giữ nguyên cho đến khi bộ định thời quay về 0
        CLR    TR0          ; Dừng Timer0
        CLR    TF0          ; Xóa cờ TF cho vòng sau.
        RET

```

Lời giải:

Để tìm giá trị cho TH ở chế độ 2 thì trình hợp ngữ cần thực hiện chuyển đổi số âm khi ta nhập vào. Điều này cũng làm cho việc tính toán trở nên dễ dàng. Vì ta đang sử dụng 150 xung đồng hồ, nên ta có thời gian trễ cho chương trình con DELAY là $150 \times 1.085\mu s$ và tần số là $f = \frac{1}{T} = 2,048kHz$.

Để ý rằng trong nhiều tính toán thời gian trễ ta đã bỏ các xung đồng hồ liên quan đến tổng phí các lệnh trong vòng lặp. Để tính toán chính xác hơn thời gian trễ và cả tần số ta đang cần phải đưa chúng vào. Nếu ta dùng một máy hiện sóng số và ta không nhận được tần số đúng như ta tính toán thì đó là do tổng phí liên quan đến các lệnh gọi trong vòng lặp.

Trong phần này ta đã dùng bộ định thời 8051 để tạo thời gian trễ. Tuy nhiên, công dụng mạnh hơn và sáng tạo hơn của các bộ định thời này là sử dụng chúng như các bộ đếm sự kiện. Chúng ta sẽ bàn về công dụng của bộ đếm này ở phần kế tiếp.

9.2 Lập trình cho bộ đếm.

Ở phần trên đây ta đã sử dụng các bộ định thời của 8051 để tạo ra các độ trễ thời gian. Các bộ định thời này cũng có thể được dùng như các bộ đếm các sự kiện xảy ra bên ngoài 8051. Công dụng của bộ đếm/ bộ định thời như bộ đếm sự kiện sẽ được trình bày ở phần này. Chừng nào còn liên quan đến công dụng của bộ định thời như bộ đếm sự kiện thì mọi vấn đề mà ta nói về lập trình bộ định thời ở phần trước cũng được áp dụng cho việc lập trình như là một bộ đếm ngoại trừ nguồn tần số. Đối với bộ định thời/ bộ đếm khi dùng nó như bộ định thời thì nguồn tần số là tần số thạch anh của 8051. Tuy nhiên, khi nó được dùng như một bộ đếm thì nguồn xung để tăng nội dung các thanh ghi TH và TL là từ bên ngoài 8051. Ở chế độ bộ đếm, hãy lưu ý rằng các thanh ghi TMOD và TH, TL cũng giống như đối với bộ định thời được bàn ở phần trước, thậm chí chúng vẫn có cùng tên gọi. Các chế độ của các bộ định thời cũng giống nhau.

9.2.1 Bit C/T trong thanh ghi TMOD.

Xem lại phần trên đây về bit C/T trong thanh ghi TMOD ta thấy rằng nó quyết định nguồn xung đồng hồ cho bộ định thời. Nếu bit C/T = 0 thì bộ định thời nhận các xung đồng hồ từ bộ giao động thạch anh của 8051. Ngược lại, khi C/T = 1 thì bộ định thời được sử dụng như bộ đếm và nhận các xung đồng hồ từ nguồn bên ngoài của 8051. Do vậy, khi bit C/T = 1 thì bộ đếm lên, khi các xung được đưa đến chân 14 và 15. Các chân này có tên là T0 (đầu vào của bộ định thời Timer0) và T1 (đầu vào của bộ Timer1). Lưu ý rằng hai chân này thuộc về cổng P3. Trong trường hợp của bộ Timer0 khi C/T = 1 thì chân P3.4 cấp xung đồng hồ và bộ đếm tăng lên đối với mỗi xung đồng hồ đi đến từ chân này. Tương tự như vậy đối với bộ Timer1 thì khi C/T = 1 với mỗi xung đồng hồ đi đến từ P3.5 bộ đếm sẽ đếm tăng lên 1.

Bảng 9.1: Các chân cổng P3 được dùng cho Timer0 và Timer1.

Chân	Chân cổng	Chức năng	Mô tả
14	P3.4	T0	Đầu vào ngoài của bộ đếm 0
15	P3.5	T1	Đầu vào ngoài của bộ đếm 1

Ví dụ 9.18:

Giả sử rằng xung đồng hồ được cấp tới chân T1, hãy viết chương trình cho bộ đếm 1 ở chế độ 2 để đếm các xung và hiển thị trạng thái của số đếm TL1 trên cổng P2.

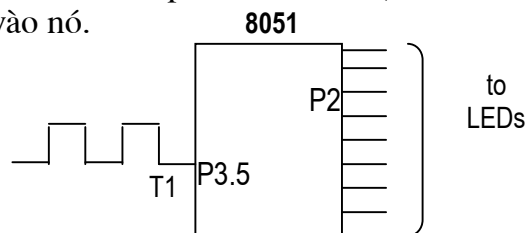
Lời giải:

```

MOV    TMOD, #01100000B    ; Chọn bộ đếm 1, chế độ 2, bit C/T = 1
                                xung ngoài.
MOV    TH1, #0              ; Xoá TH1
SETB   P3.5                 ; Lấy đầu vào T1
AGAIN: SETB   TR1            ; Khởi động bộ đếm
BACK:  MOV    A, TL1         ; Lấy bản sao số đếm TL1
        MOV    P2, A         ; Đưa TL1 hiển thị ra cổng P2.
        JNB    TF1, Back     ; Duy trì nó nếu TF = 0
        CLR    TR1          ; Dừng bộ đếm
        CLR    TF1          ; Xoá cờ TF
        SJMP   AGAIN        ; Tiếp tục thực hiện

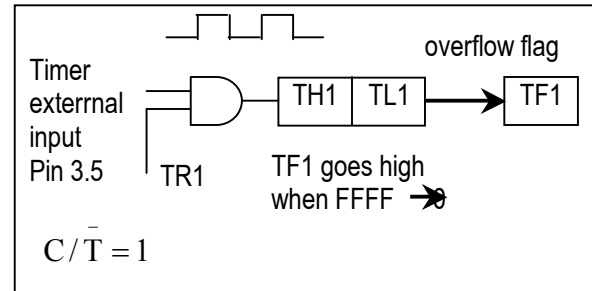
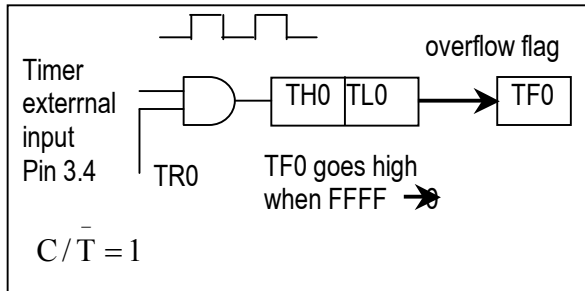
```

Để ý trong chương trình trên về vai trò của lệnh “SETB P3.5” vì các cổng được thiết lập dành cho đầu ra khi 8051 được cấp nguồn nên ta muốn P3.5 trở thành đầu vào thì phải bật nó lên cao. Hay nói cách khác là ta phải cấu hình (đưa lên cao) chân T1 (P3.5) để cho phép các xung được cấp vào nó.



Trong ví dụ 9.18 chúng ta sử dụng bộ Timer1 như bộ đếm sự kiện để nó đếm lên mỗi khi các xung đồng hồ được cấp đến chân P3.5. Các xung đồng hồ này có thể biểu diễn số người đi qua cổng hoặc số vòng quay hoặc bất kỳ sự kiện nào khác mà có thể chuyển đổi thành các xung.

Trong ví dụ 9.19 các thanh ghi TL được chuyển đổi về mã ASCII để hiển thị trên một LCD.



Hình 9.5: a) Bộ Timer0 với đầu vào ngoài (chế độ 1)

b) Bộ Timer1 với đầu vào ngoài (chế độ 1)

Ví dụ 9.19:

giả sử rằng một xung tần số 1Hz được nối tới chân đầu vào P3.4. Hãy viết chương trình hiển thị bộ đếm 0 trên một LCD. Hãy đặt số ban đầu của TH0 là - 60.

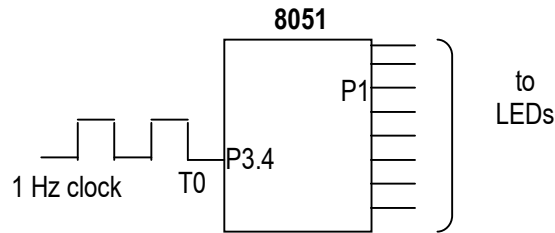
Lời giải:

Để hiển thị số đếm TL trên một LCD ta phải thực hiện chuyển đổi giữ liệu 8 bit nhị phân về ASCII.

```

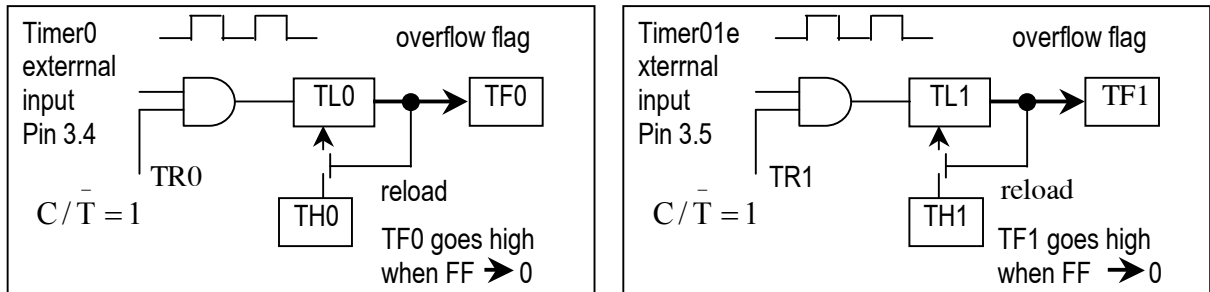
                ACALL  LCD-SET UP          ; Gọi chương trình con khởi tạo CLD
                MOV    TMOD, #000110B     ; Chọn bộ đếm 0, chế độ 2, bit C/T = 1
                MOV    TH0, # - 60         ; Đếm 60 xung
                SETB   P3.4                ; Lấy đầu vào T0
AGAIN:          SETB   TR0                 ; Sao chép số đếm TL0
BACK:           MOV    A, TL0              ; Gọi chương trình con để chuyển đổi
                                                trong các thanh ghi R2, R3, R4.
                ACALL  CONV                ; Gọi chương trình con hiển thị trên LCD
                ACALL  DISLAY              ; Thực hiện vòng lặp nếu TF = 0
                JNB    TF0, BACK            ; Dừng bộ đếm 0
                CLR    TR0                 ; Xoá cờ TF0 = 0
                CLR    TF0                 ; Tiếp tục thực hiện
                SJMP   AGAIN                ; Việc chuyển đổi nhị phân về mã ASCII
khi trả dữ liệu ASCII có trong các thanh ghi R4, R3, R2 (R2 có LSD) - chữ số nhỏ nhất.
CONV: MOV       B, #10                    ; Chia cho 10
                DIV    AB
                MOV    R2, B                ; Lưu giữ số thấp
                MOV    B, #10              ; Chia cho 10 một lần nữa
                DIV    AB
                ORL    A, #30H              ; Đổi nó về ASCII
                MOV    R4, A                ; Lưu chữ số có nghĩa lớn nhất MSD
                MOV    A, B
                ORL    A, #30H              ; Đổi số thứ hai về ASCII
                MOV    R3, A                ; Lưu nó
                MOV    A, R2
                ORL    A, #30H              ; Đổi số thứ ba về ASCII
                MOV    R2, A                ; Lưu số ASCII vào R2.
                RET

```



Sử dụng tần số 60Hz ta có thể tạo ra các giây, phút, giờ.

Lưu ý rằng trong vòng đầu tiên, nó bắt đầu từ 0 vì khi RESET thì TL0 = 0; Để giải quyết vấn đề này hãy nạp TL0 với giá trị - 60 ở đầu chương trình.



Hình 9.6: Bộ Timer0 với đầu vào ngoài (chế độ 2)

Hình 9.7: Bộ Timer1 với đầu vào ngoài (chế độ 2)

Như một ví dụ ứng dụng khác của bộ định thời gian với bit C/T = 1, ta có thể nạp một sóng vuông ngoài với tần số 60Hz vào bộ định thời. Chương trình sẽ tạo ra các đơn vị thời gian chuẩn theo giây, phút, giờ. Từ đầu vào này ta hiển thị lên một LCD. Đây sẽ là một đồng hồ số tuyệt vời nhưng nó không thật chính xác. Ví dụ này có thể tìm thấy ở phụ lục Appendix E.

Trước khi kết thúc chương này ta cần nhắc lại hai vấn đề quan trọng.

1. Chúng ta có thể nghĩ rằng công dụng của lệnh “JNB TFx, đích” để hiển thị mức cao của cờ TF là một sự lãng phí thời gian của BVĐK. Điều đó đúng có một giải pháp cho vấn đề này là sử dụng các ngắt. Khi sử dụng các ngắt ta có thể đi thực hiện các công việc khác với BVĐK. Khi cờ TF được bật thì nó báo cho ta biết đây là điểm quan trọng về thể mạnh của 8051 (mà ta sẽ bàn ở chương 11).
2. Chúng ta muốn biết các thanh ghi TR0 và TR1 thuộc về đâu. Chúng thuộc về một thanh ghi gọi là TCON mà sẽ được ban sau ở đây (TCON - là thanh ghi điều khiển bộ đếm (bộ định thời)).

Bảng 9.2: Các lệnh tương đương đối với thanh ghi điều khiển bộ định thời.

Đối với Timer0	
SETB TR0	= SETB TCON.4
CLR TR0	= CLR TCON.4
SETB TF	= SETB TCON.5
CLR TF0	= CLR TCON.5
Đối với Timer1	
SETB TR1	= SETB TCON.6

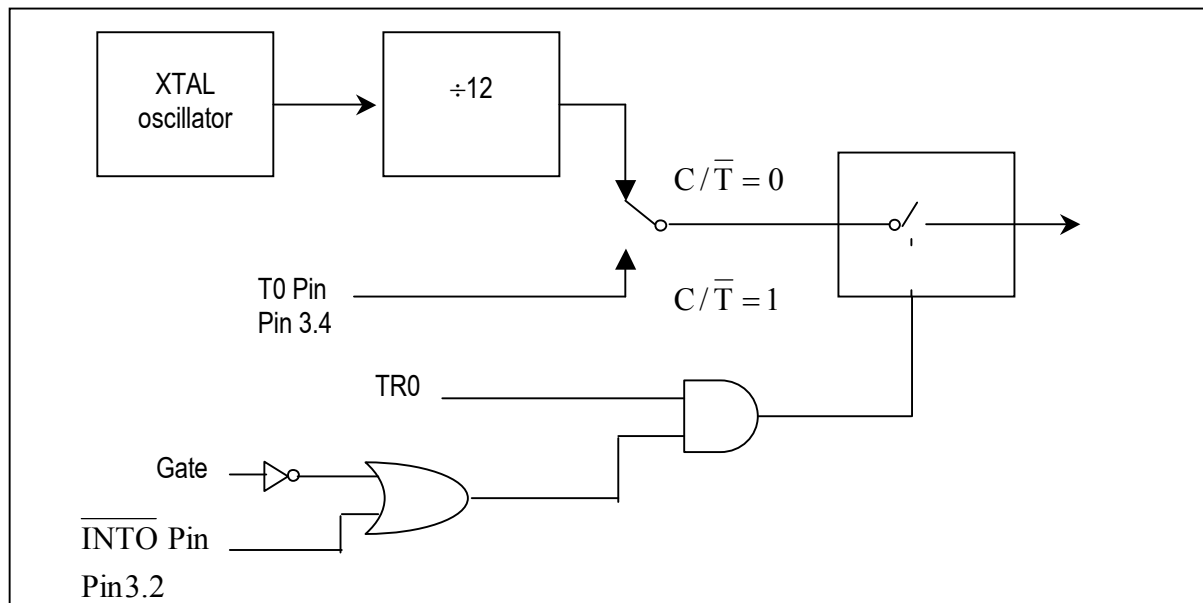
CLR TR1 = CLR TCON.6
SETB TF1 = SETB TCON.7
CLR TF1 = CLR TCON.7

9.2.2 Thanh ghi TCON.

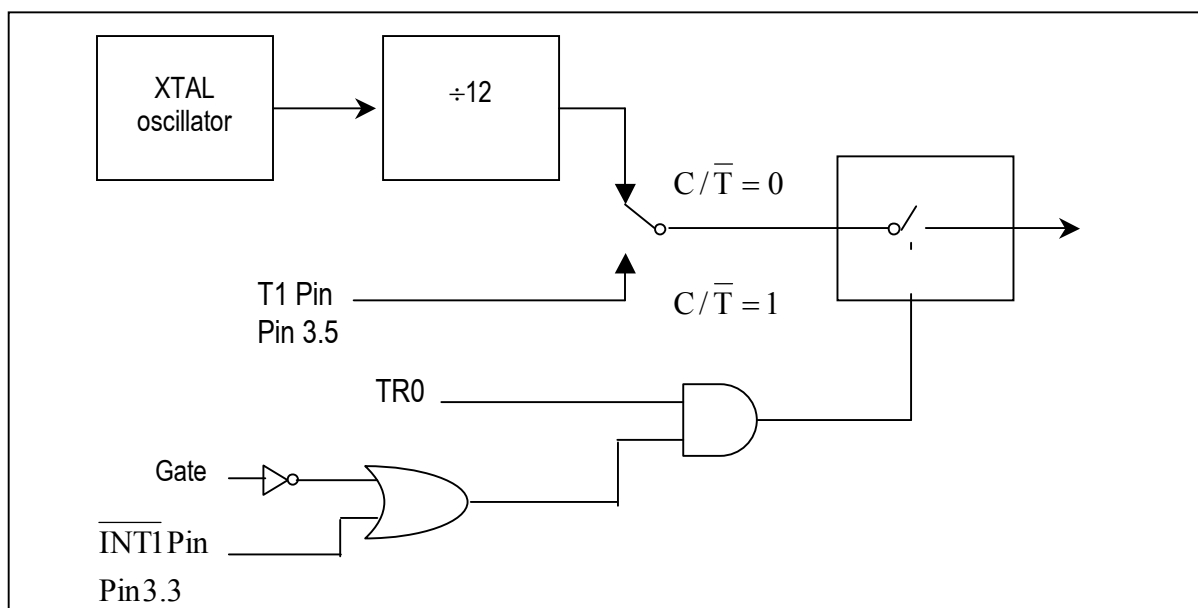
Trong các ví dụ trên đây ta đã thấy công dụng của các cờ TR0 và TR1 để bật/ tắt các bộ định thời. Các bit này là một bộ phận của thanh ghi TCON (điều khiển bộ định thời). Đây là thanh ghi 8 bit, như được chỉ ra trong bảng 9.2 thì bốn bit trên được dùng để lưu cất các bit TF và TR cho cả Timer0 và Timer1. Còn bốn bit thấp được thiết lập dành cho điều khiển các bit ngắt mà ta sẽ bàn ở chương 11. Chúng ta phải lưu ý rằng thanh ghi TCON là thanh ghi có thể đánh địa chỉ theo bit được. Nên ta có thể thay các lệnh như “SETB TR1” là “CLR TR1” bằng các lệnh tương ứng như “SET TCON.6” và “CLR TCON.6” (Bảng 9.2).

9.3 Trường hợp khi bit GATE = 1 trong TMOD.

Trước khi kết thúc chương ta cần bàn thêm về trường hợp khi bit GATE = 1 trong thanh ghi TMOD. Tất cả những gì chúng ta vừa nói trong chương này đều giả thiết GATE = 0. Khi GATE = 0 thì bộ định thời được khởi động bằng các lệnh “SETB TR0” và “SETB TR1” đối với Timer0 và Timer1 tương ứng. Vậy điều gì xảy ra khi bit GATE = 1? Như ta có thể nhìn thấy trên hình 9.8 và 9.9 thì nếu GATE = 1 thì việc khởi động và dừng bộ định thời được thực hiện từ bên ngoài qua chân P2.3 và P3.3 đối với Timer0 và Timer1 tương ứng. Mặc dù rằng TRx được bật lên bằng lệnh “SETB TRx” thì cũng cho phép ta khởi động và dừng bộ định thời từ bên ngoài tại bất kỳ thời điểm nào thông qua công tắc chuyển mạch đơn giản. Phương pháp điều khiển phân cứng để dừng và khởi động bộ định thời nay có thể có rất nhiều ứng dụng. Ví dụ, chẳng hạn 8051 được dùng trong một sản phẩm phát báo động mỗi giây dùng bộ Timer0 theo nhiều việc khác. Bộ Timer0 được bật lên bằng phần mềm qua lệnh “SETB TR0” và nằm ngoài sự kiểm soát của người dùng sản phẩm đó. Tuy nhiên, khi nối một công tắc chuyển mạch tới chân P2.3 ta có thể dừng và khởi động bộ định thời gian bằng cách đó để tắt báo động.



Hình 9.8: Bộ định thời/ bộ đếm 0.



Hình 9.9: Bộ định thời/ bộ đếm 1.