

[Home](#)[english](#)[PLC](#)[Laview](#)[lập trình C/C++](#)[thủ thuật IT](#)**Wed hay**ĐHKTCN Thái
Nguyễn[Facebook](#)[Bongdaplus](#)[Manchester Unit](#)[Dân trí](#)[24h](#)**Điện tử Anlog**

điện trở

Tụ Điện

Quận cảm

điốt

transistor

mosfet

IC khuếch đại thuật

toán

khuếch đại(K)

Tryritor(SCR)

Triac

Mạch điện thú vị

Điện tử số**Vi Điều Khiển**[giới thiệu](#)[8051](#)[VĐK 8051](#)[giới
thiệu về
họ vi
điều
khiển
8051](#)[Hướng
Dẫn Sử
Dụng
Keil C
Lập
Trình
8051
Các
chân,
cổng
vào/ra](#)[Bộ
đếm/
bộ định
thời
trong
8051](#)[Truyền
thông
nối tiếp
với
8051](#)[Ngắt
trong
8051](#)[datasheet
8051](#)[Code
tham khảo](#)[HQ pic](#)**Chuyên Ngành
(ddk)**[Home](#) > [giới thiệu](#) > [8051](#) > [VĐK 8051](#) >

Bộ đếm/ bộ định thời trong 8051

- Bộ đếm, bộ định thời là gì?
- Các thanh ghi liên quan
- Cách thức hoạt động của bộ đếm/bộ định thời
- Các bước lập trình bộ đếm/bộ định thời

Giới thiệu

Bộ đếm/Bộ định thời: Đây là các ngoại vi được thiết kế để thực hiện một nhiệm vụ đơn giản: đếm các xung nhịp. Mỗi khi có thêm một xung nhịp tại đầu vào đếm thì giá trị của bộ đếm sẽ được tăng lên 01 đơn vị (trong chế độ đếm tiến/đếm lên) hay giảm đi 01 đơn vị (trong chế độ đếm lùi/đếm xuống).

Xung nhịp đưa vào đếm có thể là một trong hai loại:

- Xung nhịp bên trong IC: Đó là xung nhịp được tạo ra nhờ kết hợp mạch dao động bên trong IC và các linh kiện phụ bên ngoài nối với IC. Trong trường hợp sử dụng xung nhịp loại này, người ta gọi là các **bộ định thời (timers)**. Do xung nhịp bên loại này thường đều đặn nên ta có thể dùng để **đếm thời gian** một cách khá chính xác.
- Xung nhịp bên ngoài IC: Đó là các tín hiệu logic thay đổi liên tục giữa 0-1 và không nhất thiết phải là đều đặn. Trong trường hợp này người ta gọi là các **bộ đếm (counters)**. Ứng dụng phổ biến của các bộ đếm là **đếm các sự kiện bên ngoài** như đếm các sản phẩm chạy trên băng chuyền, đếm xe ra/vào kho bãi...

Một khái niệm quan trọng cần phải nói đến là sự kiện “**tràn**” (**overflow**). Nó được hiểu là sự kiện bộ đếm đếm vượt quá giá trị tối đa mà nó có thể biểu diễn và quay trở về giá trị 0. Với bộ đếm 8 bit, giá trị tối đa là 255 (tương đương với FF trong hệ Hexa) và là 65535 (FFFFH) với bộ đếm 16 bit.

8051 có 02 bộ đếm/bộ định thời. 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 VĐK. Trong bài này chúng ta sẽ tìm hiểu về cách lập trình cho chúng và sử dụng chúng như thế nào. Phần 1 là Lập trình bộ định thời, và phần 2 là Lập trình cho bộ đếm.

1. Các bộ định thời của 8051

8051 có hai bộ định thời là **Timer 0** và **Timer 1**, 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.

1.1 Các thanh ghi cơ sở của bộ định thời

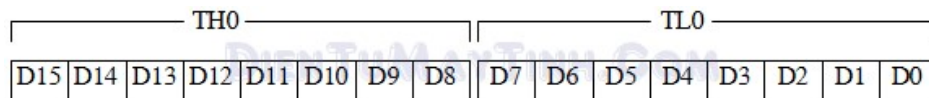
Cả hai bộ định thời **Timer 0** và **Timer 1** đều có độ dài **16 bit** được truy cập như hai thanh ghi tách biệt **byte thấp** và **byte cao**. Chúng ta sẽ bàn riêng về từng thanh ghi.

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** (Timer0 Low byte).
- Thanh ghi byte cao được gọi là **TH0** (Timer0 High byte).

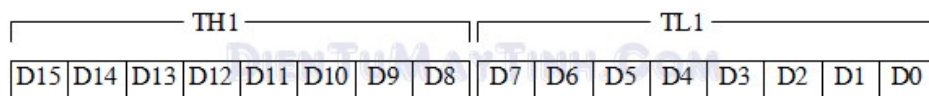
Các thanh ghi này có thể được truy cập, hoặc được đọc như mọi thanh ghi khác chẳng hạn như A, B, R0, R1, R2 v.v...



Hình 1: Các thanh ghi của bộ Timer 0

1.1.2 Các thanh ghi của bộ Timer 1

Giống như **timer 0**, bộ định thời gian **Timer 1** cũng dài 16 bit và thanh ghi 16 bit của nó cũng 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 2: Các thanh ghi của bộ Timer 1.

1.1.3 Thanh ghi TMOD

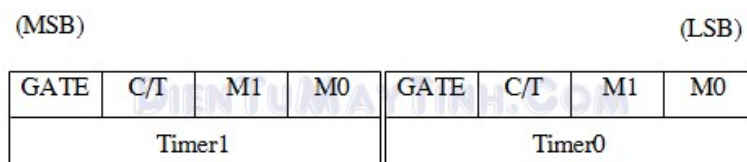
Cả hai bộ định thời **Timer 0** và **Timer 1** đều dùng chung một thanh ghi được gọi là **TMOD**: để thiết lập các **chế độ làm việc khác nhau của bộ định thời**.

Thanh ghi **TMOD** là **thanh ghi 8 bit** gồm có:

- 4 bit thấp để thiết lập cho bộ **Timer 0**.
- 4 bit cao để thiết lập cho **Timer 1**.

Trong đó:

- 2 bit thấp của chúng dùng để thiết lập chế độ của bộ định thời.
- 2 bit cao dùng để xác định phép toán.



Hình 3: Thanh ghi TMOD.

1.1.3.1 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** như **bảng dưới**. Chúng ta chỉ tập chung vào các chế độ thường được sử dụng rộng rãi nhất là **chế độ 1** và **chế độ 2**. Chúng ta sẽ sớm khám phá ra các đặc tính của các chế độ này sau khi khám phá 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: 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.

Bảng 1: Các chế độ hoạt động của bộ đếm/bộ định thời

1.1.3.2 Bit C/T (Counter/Timer)

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.

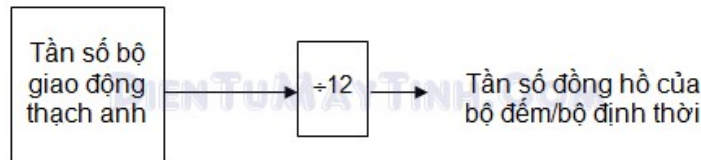
Ví dụ 1:

TMOD = 0000 0001 (01H) : chế độ 1 của bộ định thời Timer 0 được chọn.

TMOD = 0010 0000 (20H) : chế độ 2 của bộ định thời Timer 1 được chọn.

TMOD = 0001 0010 (12H) : chế độ 1 của bộ định thời Timer 1 và chế độ 2 của Timer 0 được chọn.

Nguồn đồng hồ cho chế độ trễ thời gian là tần số thạch anh của 8051. 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.



Hình 4: Tần số của bộ đếm/bộ định thời

Ví dụ 2:

Tần số thạch anh	Tần số bộ định thời	Chu kỳ bộ định thời
20MHz	$20\text{MHz}/12=1,6666\text{MHz}$	$1/1,6666\text{MHz}=0,6\mu\text{s}$
12MHz	$12\text{MHz}/12=1\text{MHz}$	$1/1\text{MHz}=1\mu\text{s}$
11,0592MHz	$11,0592\text{MHz}/12=0,9216\text{MHz}$	$1/0,9216\text{MHz}=1,085\mu\text{s}$

Bảng 2: Một số tần số thông dụng

Mặc dù các hệ thống 8051 có thể sử dụng tần số thạch anh từ 10 đến 40MHz, song ta chỉ tập trung vào tần số thạch anh **11,0592MHz**. Lý do đằng sau một số lẻ như vậy là tốc độ baud đối với truyền thông nối tiếp của 8051. Tần số XTAL = 11,0592MHz cho phép hệ thống 8051 truyền thông với PC mà không có lỗi.

1.1.3.3 Bit cổng GATE

Một bit khác của thanh ghi **TMOD** là bit cổng **GATE**. Để ý trên **hình 3** ta thấy cả hai bộ định thời Timer0 và Timer1 đều có bit **GATE**. Vậy bit **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 bit **khởi động bộ định thời TR** là **TR0** và **TR1**. Điều này có được nhờ các lệnh Set bit **TR0** lên 1 (khởi động bộ định thời) hoặc Clear bit **TR0** (dừng bộ định thời) đối với **Timer 0**, và tương tự **TR1** đối với **Timer 1**. Các lệnh này có tác dụng khi bit **GATE = 0** trong thanh ghi **TMOD**.
- Việc khởi động và ngừng bộ định thời bằng **phần cứng từ nguồn ngoài** bằng cách đặt bit **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.

Ví dụ 3:

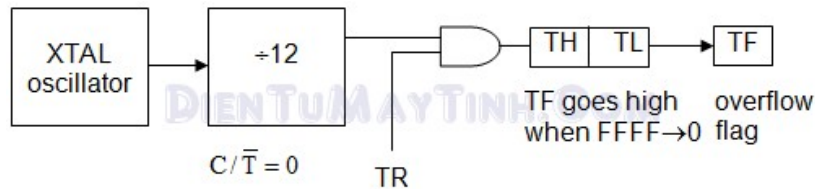
TMOD = 0000 0010: Bộ định thời là Timer0, chế độ 2, **C/T = 0** dùng nguồn XTAL, **GATE = 0** dùng phần mềm để khởi động và dừng bộ định thời.

Như vậy, bây giờ chúng ta đã có hiểu biết cơ bản về vai trò của thanh ghi **TMOD**, chúng ta sẽ xét từng chế độ của bộ định thời và cách chúng được lập trình như thế nào để tạo ra một độ trễ thời gian.

1.2 Lập trình cho chế độ 1

Dưới đây là những bước hoạt động của timer ở chế độ 1:

- Đây 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.
- 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 việc SET bit **TR0** đối với **Timer 0** và SET bit **TR1** đối với **Timer 1**.
- 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. Sau đó, khi nó quay 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, để **dừng bộ định thời**: ta thực hiện xóa các bit **TR0** đối với **Timer 0** hoặc **TR1** đối với **Timer 1**. Ở đây cũng cần phải nhắc lại là đối với mỗi bộ định thời đều có cờ **TF** riêng của mình: **TF0** đối với **Timer 0** và **TF1** đối với **Timer 1**.
- Sau khi bộ định thời đạt được giới hạn của nó là 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 cờ **TF** phải được xóa về 0.



Hình 5: Timer/counter chế độ 1

1.2.1 Các bước lập trình ở chế độ 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 kiểm tra cờ bộ định thời TF** bằng một vòng lặp để xem nó được bật lên 1 không. Thoát vòng lặp khi **TF** được lên cao.
5. **Dừng** bộ định thời.
6. **Xóa cờ TF** cho vòng kế tiếp.
7. Quay trở lại bước 2 để **nạp lại TL và TH**.

Công thức tính toán độ trễ sử dụng **chế độ 1** (16 bit) của bộ định thời đối với tần số thạch anh XTAL = f (MHz):

a) Tính theo số Hex	b) Tính theo số thập phân
$(FFFF - YYXX + 1) * 12 / f (\mu s)$ 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 $(65536 - NNNNN) * 12 / f (\mu s)$.

Bảng 3: Công thức tính độ trễ thời gian theo tần số XTAL (f)

Ví dụ 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:

```

#include<at89x51.h>           //khai báo thư viện cho VDK 89x51
void delay(void);             //khi báo nguyên mẫu hàm con tạo trễ
main()
{
    P1_5=1;                   //khởi tạo chân P1_5 ở mức cao
    while(1)                  //vòng lặp vô hạn
    {
        delay();              //chương trình con tạo trễ
        P1_5=~P1_5;           //đảo tín hiệu chân P1_5
    }
}
void delay(void)               //định nghĩa hàm delay
{
    TMOD=0x01;                //chọn timer0, chế độ 1, 16Bit
    TL0=0xF2;                  // nạp giá trị cho TL0
    TH0=0xFF;                  // nạp giá trị cho TH0
    TR0=1;                     //khởi động timer0
    while(!TF0){}              //vòng lặp kiểm tra cờ TF0
    TR0=0;                     //ngừng timer0
    TF0=0;                     //xóa cờ TF0
}

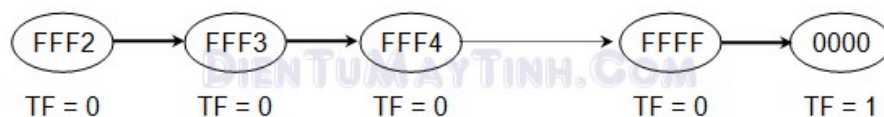
```

Trong chương trình chính (hàm main) thực hiện gọi hàm con **delay()** tạo trễ, và đảo liên tục tín hiệu đầu ra ở chân P1_5.

Trong chương trình con **delay()** trên đây chú ý các bước sau:

1. TMOD được nạp.
2. Giá trị FFF2H được nạp và TH0 - TL0
3. Bộ định thời Timer0 được khởi động bởi lệnh Set bit TR0.
4. Bộ Timer0 đếm lên 01 sau mỗi chu kỳ của timer. Khi bộ định thời đếm tăng qua các trạng thái FFF3, FFF4 ... cho đến khi đạt giá trị FFFFH là nó quay về 0000H và bật cờ bộ định thời TF0 = 1. Tại thời điểm này vòng lặp kiểm tra cờ TF0 mới được thoát ra.
5. Bộ Timer0 được dừng bởi lệnh clear bit TR0.
6. Cờ TF0 cũng được xóa, sẵn sàng cho chu trình tiếp theo.

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 (đơn giản là ta gọi lại hàm delay()).



Hình 6: Một chu trình đếm của timer0

Tính toán độ trễ tạo ra bởi bộ định thời ở chương trình trên với tần số XTAL=11,0592MHz:

Bộ định thời làm việc với tần số đồng hồ bằng 1/12 tần số XTAL, do vậy ta có $11,0592\text{MHz}/12=0,9216\text{MHz}$ là tần số của bộ định thời. Kết quả là mỗi nhịp xung đồng hồ có chu kỳ $T=1/0,9216\text{MHz}=1,085\mu\text{s}$. Hay nói cách khác, bộ Timer0 tăng 01 đơn vị sau $1,085\mu\text{s}$ để tạo ra bộ trễ bằng $\text{số_đếm} \times 1,085\mu\text{s}$.

Số đếm bằng $\text{FFFFH} - \text{FFF2H} = \text{ODH}$ (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 0000H 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.

Tuy nhiên, trong tính toán độ trễ ở trên ta đã không tính đến tổng phí các lệnh cài đặt timer0, các lệnh kiểm tra trong vòng lặp, gọi hàm con... Chính các câu lệnh này làm cho độ trễ dài hơn, dẫn đến tần số của xung vuông ở đầu ra P1_5 không còn đúng như tính toán ở trên. Đây là nhược điểm của C trong lập trình VDK. Tùy vào từng chương trình biên dịch, mỗi lệnh của C sẽ được biên dịch ra số lệnh ASM khác nhau, để tính toán chính xác ta phải tính cả tổng phí từng dòng lệnh ASM.

1.2.2 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 thống 8051.

Các bước để tìm các giá trị của các thanh ghi **TH** và **TL**:

1. Chia thời gian trễ cần thiết cho $1.085\mu s$
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: ta có YYXX là giá trị Hexa ban đầu cần phải nạp vào các thanh ghi bộ định thời.
4. Đặt TL = XX và TH = YY.

Ví dụ 5:

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 5ms.

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 5ms. 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.

```
void delay(void)           //định nghĩa hàm delay
{
    TMOD=0x01;           //chọn timer0 chế độ 1 16Bit
    TL0=0x00;             //nạp giá trị cho TL0
    TH0=0xEE;             //nạp giá trị cho TH0
    TR0=1;                //khởi động timer0
    while(!TF0){}         //vòng lặp kiểm tra cờ TF0
    TR0=0;                //ngừng timer0
    TF0=0;                //xóa cờ TF0
}
```

Ví dụ 6:

Giả sử ta có tần số XTAL là 11,0592MHz. Hãy tìm các giá trị cần được nạp vào các thanh ghi TH và TL để tạo ra một sóng vuông tần số 2kHz.

Xét các bước sau:

1. $T = 1/f = 1/2\text{KHz} = 500\mu s$ là chu kỳ của sóng vuông.
2. Khoảng thời gian phần cao và phần thấp là: $T/2 = 250\mu s$.
3. Số nhịp cần trong thời gian đó là: $250\mu s / 1.085\mu s = 230$. Giá trị cần nạp vào các thanh ghi cần tìm là $65536 - 230 = 65306$ và ở dạng hex là FF1AH.
4. Giá trị nạp vào TL là 1AH, TH là FFH.

Chương trình cần viết là:

```
void delay(void)           //định nghĩa hàm delay
{
    TMOD=0x10;           //chọn timer1 chế độ 1 16Bit
    TL1=0x1A;            //nạp giá trị cho TL1
    TH1=0xFF;            //nạp giá trị cho TH1
    TR1=1;               //khởi động timer1
    while(!TF1){}        //vòng lặp kiểm tra cờ TF1
    TR1=0;               //ngừng timer1
    TF1=0;               //xóa cờ TF1
}
```

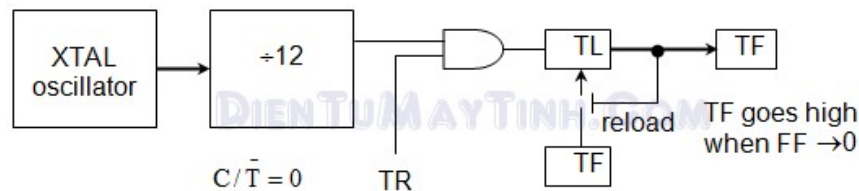
1.3 Chế độ 0

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 **1FFFF** thì nó sẽ quay trở về **0000** và cờ **TF** được bật lên.

1.4 Lập trình cho chế độ 2

Dưới đây là những bước hoạt động của timer ở chế độ 2:

- 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.
- Sau khi 2 thanh ghi **TH** và **TL** được nạp giá trị ban đầu thì bộ định thời phải được **khởi động**.
- 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**.
- Khi thanh ghi **TL** quay trở về **00** từ **FFH**, cờ **TF** được bật lên 1 thì thanh ghi **TL** được tự động nạp lại với giá trị sao chép từ 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 tự nạp lại** so với chế độ 1 (phải nạp lại các thanh ghi **TH** và **TL**).



Hình 7: Timer/counter chế độ 2

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 **giá trị ban đầu của TH được giữ nguyên, còn TL được nạp lại giá trị sao chép từ TH**.

Chế độ này có nhiều ứng dụng, bao gồm việc **thiết lập tần số baud trong truyền thông nối tiếp**.

1.4.1 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 thanh ghi **TH** và **TL** với giá trị đếm ban đầu.
3. **Khởi động** bộ định thời.
4. **Duy trì kiểm tra cờ bộ định thời TF** bằng cách sử dụng một vòng lặp để xem nó đã được bật chưa. Thoát vòng lặp khi **TF** lên cao.
5. **Dừng bộ định thời**.
6. **Xoá cờ TF**.
7. **Quay trở lại bước 3**. Vì chế độ 2 là chế độ tự nạp lại.

Ví dụ 7 minh họa những điều này:

Ví dụ 7:

```
#include<at89x51.h>           //khai báo thư viện cho VDK 89x51
void delay(void);              //khi báo nguyên mẫu hàm con tạo trễ
main()
{
    TMOD=0x20;                 //chọn timer1, chế độ 2, 8Bit, tự nạp lại
    TH1=0x00;                  //nạp giá trị cho TH1
    TL1=0xFE;                   //nạp giá trị cho TL1
```

```

P1_5=1;           //khởi tạo chân P1_5 ở mức cao
while(1)          //vòng lặp vô hạn
{
    delay();       //gọi chương trình con tạo trễ
    P1_5=~P1_5;    //đảo tín hiệu chân P1_5
}
}
void delay(void)   //định nghĩa hàm delay
{
    TR1=1;         //khởi động timer1
    while(!TF1){}  //vòng lặp kiểm tra cờ TF1
    TR1=0;         //ngừng timer1
    TF1=0;         //xóa cờ TF1
}

```

Hàm delay() trên sẽ tạo một độ trễ bằng 256 lần ($FF - 00 + 1$) chu kỳ của timer (không tính tổng phí các lệnh) kể từ chu trình thứ 2. Vì chu trình đầu tiên timer1 bắt đầu đếm ở vị trí **0xFE**, kể từ chu trình sau thì thanh ghi **TL1** mới sao chép được giá trị ở **TH1**.

2. 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 (counter)** các sự kiện xảy ra bên ngoài 8051. Công dụng của 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 đếm cũng giống nhau.

2.1 Bit C/T trong thanh ghi TMOD

Xem lại phần trên về bit **C/T** trong thanh ghi **TMOD**: ta thấy rằng nó quyết định nguồn xung đồng hồ cho bộ đếm:

- Nếu bit **C/T = 0** thì bộ định thời nhận các xung đồng hồ từ bộ giao động thạch anh của 8051.
- Nếu bit **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, nếu bit **C/T = 1** thì bộ đếm tăng lên khi các xung được đưa đến chân P3.4 (**T0**) đối với **counter0** và chân P3.5 (**T1**) đối với **counter1**.

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

Bảng 4: Các chân cổng P3 được dùng cho bộ đếm 0 và 1

Ví dụ 8:

Chương trình sau sử dụng bộ đếm 1, đếm các xung ở chân P3.5 và hiển thị số đếm được (trong thanh ghi TL1) lên cổng P2:

```

#include<at89x51.h>    //khai báo thư viện 89x51
main()                //chương trình chính
{

```



```

TMOD=0x60; //0x60=0110 000 : C/T=1, bộ đếm 1, chế độ 2 tự nạp
TH1=0x00; //xóa bộ đếm ban đầu

P3_5=1; //set chân vào cho bộ đếm
TR1=1; //khởi động bộ đếm 1

while(1) //vòng lặp vô hạn
{
    P2=TL1; //hiển thị số đếm được ra cổng P2
}
}

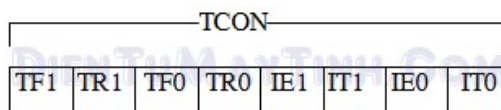
```

Trong **ví dụ 8** chúng ta sử dụng bộ **counter1** 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.

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 đếm/bộ định thời. Các bit này là một bộ phận của thanh ghi **TCON**. Đây là thanh ghi **8 bit**, như được chỉ ra trong **hình 2**:

- 4 bit trên được dùng để lưu cất các bit **TF** và **TR** cho cả Timer/counter 0 và Timer/counter 1.
- 4 bit thấp được thiết lập dành cho điều khiển các **ngắt** mà ta sẽ bàn ở các bài sau.



Hình 8: Thanh ghi TCON – Điều khiển bộ đếm/bộ định thời

2.3 Trường hợp khi bit GATE = 1 trong TMOD

Trước khi kết thúc bài này ta cần bàn thêm về trường hợp khi bit **GATE = 1** trong thanh ghi **TMOD**. Tất cả những gì chúng ta vừa nói trong bài này đều giả thiết **GATE = 0**. Khi **GATE = 0** thì bộ đếm/bộ định thời được khởi động bằng các lệnh Set bit **TR0** hoặc **TR1**. Vậy điều gì xảy ra khi bit **GATE = 1**?

Nếu **GATE = 1** thì việc khởi động và dừng bộ đếm/bộ định thời được thực hiện từ bên ngoài qua chân **P3.2 (INT0)** và **P3.3 (INT1)** đối với Timer/counter 0 và Timer/counter 1 tương ứng. Phương pháp điều khiển bằng phần cứng để dừng và khởi động bộ đếm/bộ định thời này 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 Set bit **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, bằng cách đó ta có thể tắt báo động.

Nhận xét

Bạn không có quyền thêm nhận xét.