Chương 3: CÁC HOẠT ĐỘNG CỦA VI ĐIỀU KHIỂN MCS-51

Chương này giới thiệu về các hoạt động đặc trưng của họ vi điều khiển MCS-51: định thời, cổng nối tiếp, ngắt và các cách thức để điều khiển các hoạt động này.

1. Hoạt động định thời (Timer / Counter)

1.1. Giới thiệu

AT89C51 có 2 bộ định thời 16 bit có thể hoạt động ở các chế độ khác nhau và có khả năng định thời hay đếm sự kiện (Timer 0 và Timer 1). Khi hoạt động định thời (timer), bộ Timer / Counter sẽ nhận xung đếm từ dao động nội còn khi đếm sự kiện (counter), bộ Timer / Counter nhận xung đếm từ bên ngoài. Bộ Timer / Counter bên trong AT89C51 là các bộ đếm lên 8 bit hay 16 bit tuỳ theo chế độ hoạt động. Mỗi bộ Timer / Counter có 4 chế độ hoạt động khác nhau và được dùng để:

- Đếm sự kiện tại các chân T0 (chân 14) hay T1 (chân 15).
- Chờ một khoảng thời gian.
- Tạo tốc độ cho port nối tiếp.

Quá trình điều khiển hoạt động của Timer / Counter được thực hiện thông qua các thanh ghi sau:

Bảng 3.1 – Các thanh ghi điều khiển hoạt động Timer / Counter

Thanh ghi	Địa chỉ byte	Địa chỉ bit
TCON	88h	88h – 8Fh
TMOD	89h	Không
TL0	90h	Không
TL1	91h	Không
TH0	92h	Không
TH1	93h	Không

Ngoài ra, trong họ 8x52 còn có thêm bộ định thời thứ 3 (Timer 2).

1.2. Hoạt động Timer / Counter

Hoạt động cơ bản của Timer / Counter gồm có các thanh ghi timer THx và TLx (x = 0, 1) mắc liên tầng tạo thành dạng thanh ghi 16 bit. Khi set bit TRx trong thanh ghi TCON (xem thêm phần 1.3), timer tương ứng sẽ hoạt động và giá trị trong thanh ghi TLx tăng lên 1 sau mỗi xung đếm. Khi TLx tràn (thay đổi từ 255 → 0), giá trị của THx tăng lên 1. Khi THx tràn, cờ tràn tương ứng TFx (trong thanh ghi TCON) sẽ được đưa lên mức 1.

Tuỳ theo nội dung của bit $C/\overline{T}$ (xem thêm thanh ghi TMOD, phần 1.3), xung đếm có thể lấy từ dao động nội ($C/\overline{T}$ = 0) hay từ các chân Tx bên ngoài ($C/\overline{T}$ = 1). Lưu ý rằng phải xoá bit TRx khi thay đổi chế độ hoạt động của Timer.

Khi xung đếm lấy từ dao động nội, tốc độ đếm = $f_{OSC}/12$ hay $f_{OSC}/2$ trong chế độ X2(nghĩa là nếu f_{OSC} = 12 MHz thì tốc độ xung đếm là 1 MHz hay cứ 1 µs thì có 1 xung đếm trong chế dộ chuẩn) hay tốc độ đếm = $f_{PER}/6$ (f_{PER}: tần số xung ngoại vi – peripheral clock).

Khi lấy xung đếm từ bên ngoài (các chân Tx),bộ đếm sẽ tăng lên 1 khi ngõ vào Tx ở mức 1 trong 1 chu kỳ và xuống mức 0 trong chu kỳ kế tiếp. Do đó, tần số xung tối đa tại các chân Tx là $f_{OSC}/24$ trong chế độ thường hay $f_{OSC}/12$ trong chế độ X2 (=$f_{PER}/12$).

1.3. Các thanh ghi điều khiển hoạt động

1.3.1. Thanh ghi điều khiển timer (TCON – Timer/Counter Control Register)

TCON chứa các bit trạng thái và các bit điều khiển cho Timer 1, Timer 0.

Bảng 3.2 – Nội dung thanh ghi TCON

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit	Ký hiệu	Địa chỉ	Mô tả
TCON.7	TF1	8Fh	Cờ báo tràn timer 1 (Timer 1 overflow Flag). Được xoá bởi phần cứng khi chuyển đến chương trình con xử lý ngắt hay xoá bằng phần mềm. Đặt bằng phần cứng khi Timer 1 tràn
TCON.6	TR1	8Eh	Điều khiển Timer 1 chạy (Timer 1 Run Control Bit). Cho phép Timer 1 hoạt động (= 1) hay ngừng (= 0).
TCON.5	TF0	8Dh	Timer 0 overflow Flag
TCON.4	TR0	8Ch	Timer 0 Run Control Bit
TCON.3	IE1	8Bh	Dùng cho ngắt ngoài 0 và 1 (sẽ xét trong phần 3 – xử lý ngắt)
TCON.2	IT1	8Ah	
TCON.1	IE0	89h	
TCON.0	IT0	88h	

Giá trị khi reset: TCON = 00h

1.3.2. Thanh ghi chế độ timer (TMOD – Timer/Counter Mode)

Thanh ghi TMOD chứa hai nhóm 4 bit dùng để đặt chế độ làm việc cho Timer 0, và Timer 1. Lưu ý rằng khi lập trình cho AT89C51, thông thường thanh ghi TMOD chỉ được gán một lần ở đầu chương trình.

Bảng 3.3 – Nội dung thanh ghi TMOD

GATE1	$C/\overline{T}1$	M11	M01	GATE0	$C/\overline{T}0$	M10	M00

<table>
<tr><th>Bit</th><th>Tên</th><th>Timer</th><th>Mô tả</th><th>Timer</th></tr>
<tr><td>7</td><td>GATE1</td><td>1</td><td>Timer 1 Gating Control Bit
GATE = 0: timer hoạt động bình thường
GATE = 1: timer chỉ hoạt động khi chân $\overline{INT1}$ = 1</td><td rowspan="5">Dùng cho Timer 1</td></tr>
<tr><td>6</td><td>C/T1</td><td>1</td><td>Timer 1 Timer/Counter Select Bit
= 1: đếm bằng xung ngoài tại chân T1 (chân 15)
= 0: đếm bằng xung dao động bên trong</td></tr>
<tr><td>5</td><td>M11</td><td>1</td><td rowspan="3">Timer 1 Mode Select Bit
<table>
<tr><th>M11</th><th>M01</th><th>Chế độ</th></tr>
<tr><td>0</td><td>0</td><td>13 bit</td></tr>
<tr><td>0</td><td>1</td><td>8 bit tự động nạp lại</td></tr>
<tr><td>1</td><td>0</td><td>16 bit</td></tr>
<tr><td>1</td><td>1</td><td>Không dùng Timer 1</td></tr>
</table></td></tr>
<tr><td>4</td><td>M01</td><td>1</td></tr>
<tr><td>3</td><td>GATE0</td><td>0</td><td>Timer 0 Gating Control Bit</td><td rowspan="4">Dùng cho Timer 0</td></tr>
<tr><td>2</td><td>C/T0</td><td>0</td><td>Timer 0 Timer/Counter Select Bit</td></tr>
<tr><td>1</td><td>M10</td><td>0</td><td rowspan="2">Timer 0 Mode Select Bit
Các chế độ giống như timer 1 trong đó chế độ 3 dùng TH0 và TL0 làm 2 giá trị đếm của timer 0 và timer 1 (xem thêm phần 1.4)</td></tr>
<tr><td>0</td><td>M00</td><td>0</td></tr>
</table>

Giá trị khi reset: TMOD = 00h

Ngoài ra, Timer còn các thanh ghi chứa giá trị đếm: TH0, TL0 (Timer 0) và TH1, TL1 (Timer 1), mỗi thanh ghi có kích thước 8 bit. Giá trị các thanh ghi này khi reset cũng là 00h.

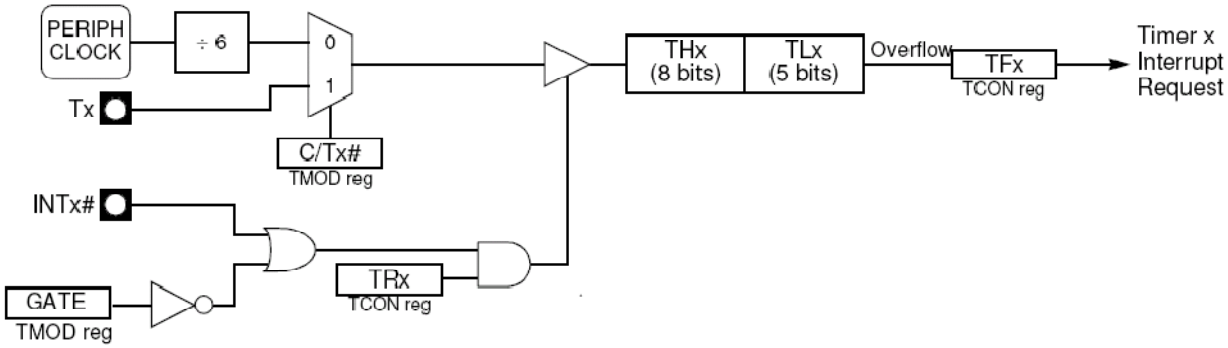
1.4. Các chế độ hoạt động

Các chế độ của timer được xác định bằng 4 bit trong thanh ghi TMOD, trong đó 4 bit thấp điều khiển timer 0 và 4 bit cao điều khiển timer 1, mô tả như sau:

1.4.1. Chế độ 0

Chế độ 0 là chế độ 13 bit bao gồm 8 bit của thanh ghi THx và 5 bit của thanh ghi TLx còn 3 bit cao của thanh ghi TLx không sử dụng. Mỗi lần có xung đếm, giá trị trong thanh ghi 13 bit tăng lên 1. Khi giá trị này thay đổi từ 1 1111 1111 1111b đến 0 thì bộ đếm tràn làm cho TFx được đặt lên mức 1.

Do chế độ 0 sử dụng 13 bit nên giá trị đếm tối đa là 2^{13} = 8192. Chế độ này được cung cấp nhằm mục đích tạo khả năng tương thích với 8048 và thường không được sử dụng hiện nay.

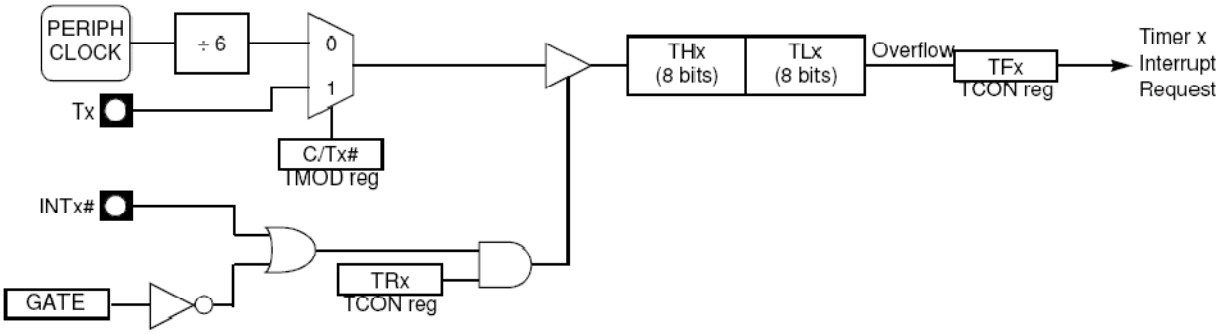


Hình 3.1 – Chế độ 0 của Timer/Counter

1.4.2. Chế độ 1

Chế độ 1 giống như chế độ 0 nhưng sử dụng 16 bit bao gồm 8 bit của THx và 8 bit của TLx nên giá trị đếm tối đa là 2^{16} = 65536. Như vậy, chế độ 0 và chế độ 1 giống nhau nhưng chỉ khác ở số bit đếm nên thông thường chế độ 0 không sử dụng mà chỉ dùng chế độ 1.

Khi bộ đếm tràn (giá trị trong cặp thanh ghi THx_TLx thay đổi từ 1111 1111 1111 1111b đến 0), cờ tràn TFx được set lên mức 1. Lưu ý rằng, khi timer tràn, giá trị của các thanh ghi đếm là 0 (THx = 0 và TLx = 0) nên nếu muốn timer hoạt động tiếp thì phải nạp lại giá trị cho các thanh ghi THx và TLx.



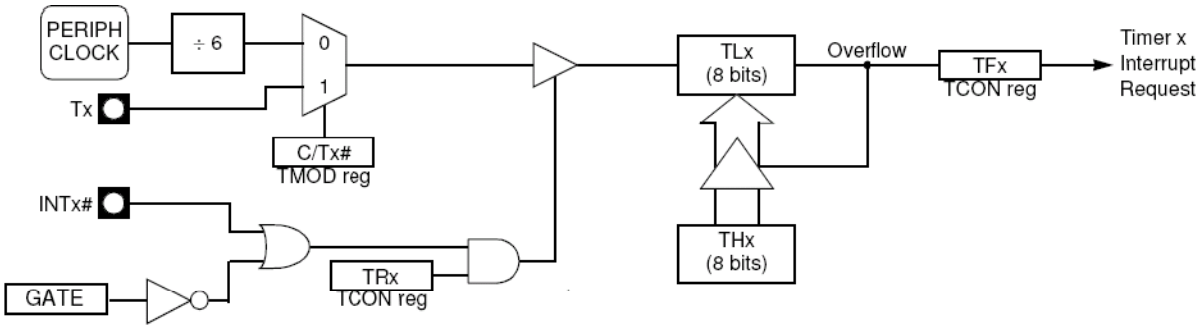
Hình 3.2 – Chế độ 1 của Timer/Counter

1.4.3. Chế độ 2

Chế độ 2 là chế độ 8 bit trong đó sử dụng thanh ghi TLx để chứa giá trị đếm còn thanh ghi THx chứa giá trị nạp lại (do đó chế độ này được gọi là chế độ tự động nạp lại – autoreload).

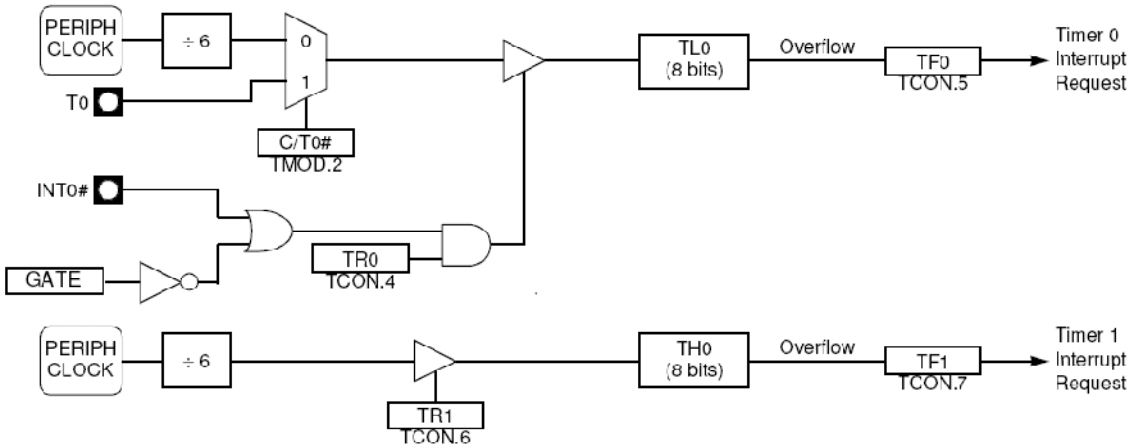
Trong chế độ 2, mỗi khi giá trị trong thanh ghi TLx thay đổi từ 1111 1111b đến 0 thì cờ TFx được set lên mức 1 đồng thời giá trị trong thanh ghi THx được chuyển vào thanh ghi TLx. Như vậy, giá trị đếm trong TLx và THx chỉ được nạp một lần khi khởi động timer (có thể không cần nạp cho TLx nhưng khi đó chu kỳ hoạt động đầu tiên của timer sẽ sai).

Chế độ 2 sử dụng 8 bit đếm trong thanh ghi TLx nên giá trị đếm tối đa là 2^8 = 256.



Hình 3.3 – Chế độ 2 của Timer/Counter

1.4.4. Chế độ 3



Hình 3.4 – Chế độ 3 của Timer/Counter

Chế độ 3 sử dụng các thanh ghi TL0 và TH0 như các bộ định thời độc lập trong đó TL0 điều khiển bằng các thanh ghi của timer 0 và TH0 điều khiển bằng các thanh ghi của tỉmer 1. Khi TL0 chuyển từ giá trị 1111 1111b đến 0 thì TF0 được đặt lên mức 1 còn TH0 chuyển từ 1111 1111b đến 0 thì TF1 được đặt lên mức 1. Lưu ý rằng trong chế độ 3 (chỉ có trong Timer 0), Timer 1 không tác động đến cờ TF1 nên thường được

dùng để tạo tốc độ baud cho port nối tiếp (xem thêm phần 2 – cổng nối tiếp) hay dùng cho mục đích khác.

Chế độ này chỉ cho phép tác động đến cờ tràn TF1 thông qua xung đếm của dao động nội mà không đếm bằng dao động ngoài tại chân T1 đồng thời bit GATE1 (TMOD.7) không tác động đến quá trình đếm tại TH0.

1.5. Timer 2

Timer 2 là bộ định thời 16 bit (chỉ có trong họ 8x52). Giá trị đếm của timer 2 chứa trong các thanh ghi TH2 và TL2. Giống như timer 0 và timer1, timer 2 cũng hoạt động như bộ định thời (timer) hay đếm sự kiện (counter). Chế độ định thời đếm bằng dao động nội, chế độ đếm sự kiện đếm bằng xung ngoài tại chân T2 (P1.0) và chọn chế độ bằng bit $C/\overline{T}2$ của thanh ghi T2CON. Các thanh ghi điều khiển timer 2 bao gồm: T2CON, T2MOD, RCAP2H, RCAP2L, TH2 và TL2.

Timer 2 có 3 chế độ hoạt động: capture (giữ), autoreload (tự động nạp lại) và tạo tốc độ baud (chọn chế độ trong thanh ghi T2CON). Các bit chọn chế độ được mô tả như bảng 3.4.

Bảng 3.4 – Chọn chế độ trong Timer 2

RCLK	TCLK	$CP/\overline{RL}2$	TR2	Chế độ
0	0	0	1	Tự động nạp lại 16 bit
0	0	1	1	Giữ 16 bit
X	1	X	1	Tạo tốc độ baud
1	X	X	1	
X	X	X	0	Ngưng

1.5.1. Các thanh ghi điều khiển Timer 2

❖ **Thanh ghi T2CON:**

Bảng 3.5 – Nội dung thanh ghi T2CON

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T}2$	$CP/\overline{RL}2$

Bit	Tên	Mô tả
7	TF2	Timer 2 overflow Flag TF2 không được tác động khi RCLK hay TCLK = 1. TF2 phải được xoá bằng phần mềm và được đặt bằng phần cứng khi Timer tràn
6	EXF2	Timer 2 External Flag Được đặt khi EXEN2 = 1 và xảy ra chế độ nạp lại hay giữ do có cạnh âm tại chân T2EX (P1.1) (chuyển từ 1 xuống 0). Khi EXF2 = 1 và cho phép ngắt tại Timer 2 thì chương trình sẽ chuyển đến chương trình phục vụ ngắt của Timer 2. EXF2 phải được xoá bằng phần mềm

5	RCLK	Receive Clock Bit (chỉ dùng cho port nối tiếp ở chế độ 1 và 3) RCLK = 0: dùng timer 1 làm xung clock thu cho port nối tiếp RCLK = 1: dùng timer 2 làm xung clock thu cho port nối tiếp
4	TCLK	Transmit Clock Bit Giống như RCLK nhưng dùng cho xung clock phát
3	EXEN2	Timer 2 External Enable Bit = 0: bỏ qua tác động tại chân T2EX (P1.1) = 1: xảy ra chế độ nạp lại hay giữ do có cạnh âm tại chân T2EX (P1.1) (chuyển từ 1 xuống 0)
2	TR2	Timer 2 Run Control Bit = 0: cấm timer 2 = 1: chạy timer 2
1	$C/\overline{T}2$	Timer / Counter 2 Select Bit = 0: định thời (đếm bằng dao động nội) = 1: đếm sự kiện (đếm bằng xung tại T2 (P1.0))
0	$CP/\overline{RL}2$	Timer 2 Capture / Reload Bit Nếu RCLK = 1 hay TCLK = 1: bỏ qua Nếu RCLK = 0 và TCLK = 0: chọn chế độ giữ (= 1) hay nạp lại (= 0) khi xuất hiện xung âm tại T2EX (P1.1) và EXEN2 = 1

Giá trị khi reset: T2CON = 00h, T2CON cho phép định vị bit

❖ **Thanh ghi T2MOD:**

Bảng 3.6 – Nội dung thanh ghi T2MOD

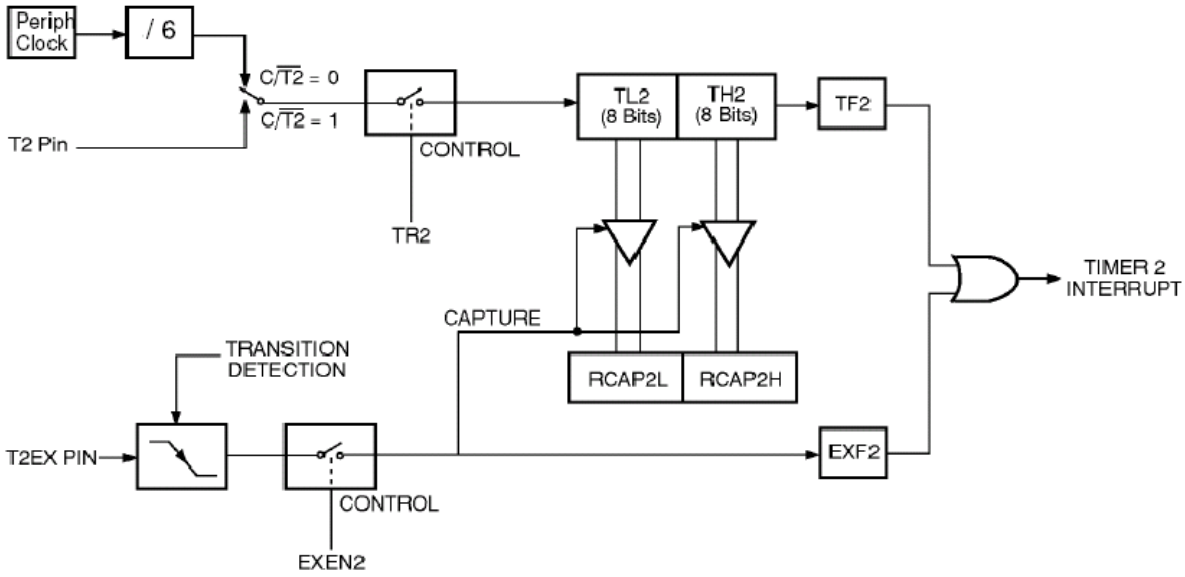
-	-	-	-	-	-	T2OE	DCEN

Bit	Tên	Mô tả
7	-	
6	-	
5	-	
4	-	
3	-	
2	-	
1	T2OE	Timer 2 Output Enable Bit = 0: T2 (P1.0) là ngõ vào clock hay I/O port = 1: T2 là ngõ ra clock
0	DCEN	Down Counter Enable Bit = 0: cấm timer 2 là bộ đếm lên / xuống = 1: cho phép timer 2 là bộ đếm lên / xuống

Giá trị khi reset: T2MOD = xxxx xx00b, MOD không cho phép định vị bit

Các thanh ghi TH2, TL2, RCAP2H và RCAP2L không cho phép định vị bit và giá trị khi reset là 00h. Các chế độ hoạt động của Timer 2 mô tả trong phần sau.

1.5.2. Chế độ capture



Hình 3.5 – Chế độ giữ của Timer 2

Chế độ giữ của Timer 2 có 2 trường hợp xảy ra:

- Nếu EXEN2 = 0: Timer 2 hoạt động giống như Timer 0 và 1, nghĩa là khi giá trị đếm tràn (TH2_TL2 thay đổi từ FFFFh đến 0) thì cờ tràn TF2 được đặt lên mức 1 và tạo ngắt tại Timer 2 (nếu cho phép ngắt).
- Nếu EXEN2 = 1: vẫn hoạt động như trên nhưng thêm một tính chất nữa là: khi xuất hiện cạnh âm tại chân T2EX (P1.1), giá trị hiện tại của TH2 và TL2 được chuyển vào cặp thanh ghi RCAP2H, RCAP2L (quá trình giữ (capture) xảy ra); đồng thời, bit EXF2 = 1 (sẽ tạo ngắt nếu cho phép ngắt tại Timer 2).

1.5.3. Chế độ tự động nạp lại

Chế độ tự động nạp lại cũng có 2 trường hợp giống như chế độ giữ:

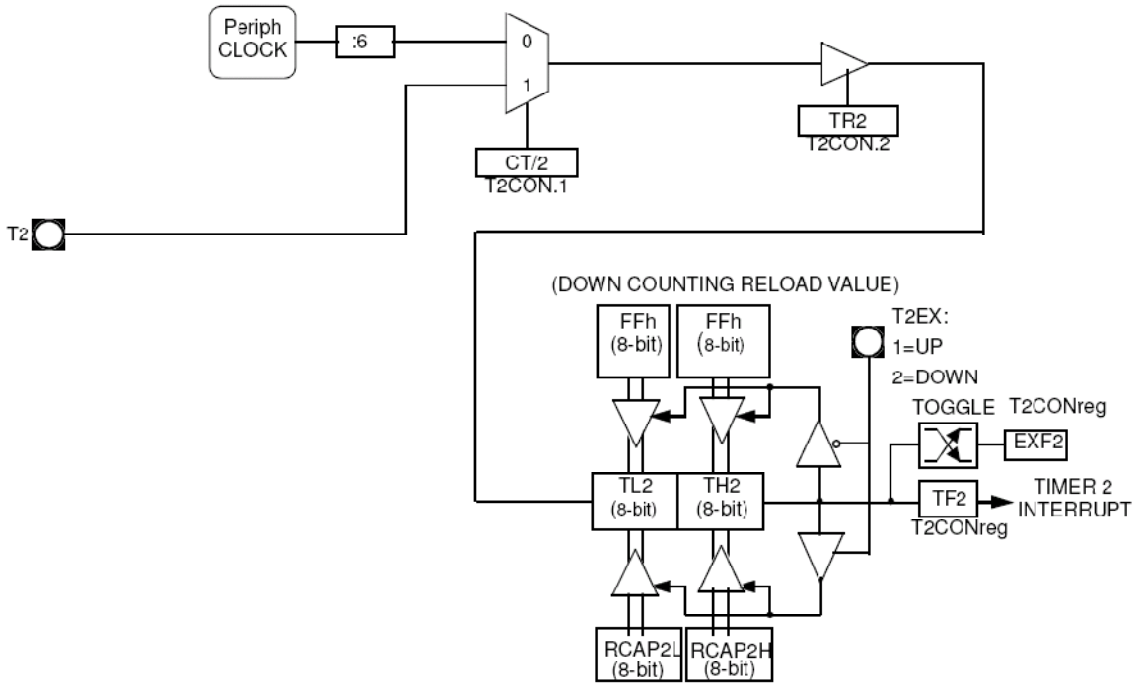
- Nếu EXEN2 = 0: khi Timer tràn, cờ tràn TF2 được đặt lên 1 và nạp lại giá trị cho TH2, TL2 (từ cặp thanh ghi RCAP2H, RCAP2L) đồng thời tạo ngắt tại timer 2 nếu cho phép ngắt.
- Nếu EXEN2 = 1: hoạt động giống như trên nhưng khi có xung âm tại chân T2EX thì cũng nạp lại giá trị cho TH2, TL2 và đặt cờ EXF2 lên 1.

Chế độ tự động nạp lại cũng cho phép thực hiện đếm lên hay xuống (điều khiển bằng bit DCEN trong thanh ghi T2MOD). Khi DCEN được đặt lên 1 và chân T2EX ở mức cao thì timer 2 sẽ đếm lên; còn nếu T2EX ở mức thấp thì timer 2 đếm xuống.

Khi đếm lên, tỉmer tràn tại giá trị đếm 0FFFFh. Khi tràn, cờ TF2 được đặt lên mức 1 và giá trị trong cặp thanh ghi RCAP2H, RCAP2L chuyển vào căp thanh ghi TH2, TL2.

Khi đếm xuống, timer tràn khi giá trị trong cặp thanh ghi TH2, TL2 bằng giá trị trong cặp thanh ghi RCAP2H, RCAP2L. Khi tràn, cờ TF2 được đặt lên 1 và giá trị 0FFFFh được nạp vào cặp thanh ghi TH2, TL2.

Trong chế độ này, khi timer tràn, giá trị trong cờ EXF2 sẽ chuyển mức và không tạo ngắt (có thể dùng thêm EXF2 để tạo giá trị đếm 17 bit).



Hình 3.6 – Chế độ tự động nạp lại

1.5.4. Chế độ tạo xung clock

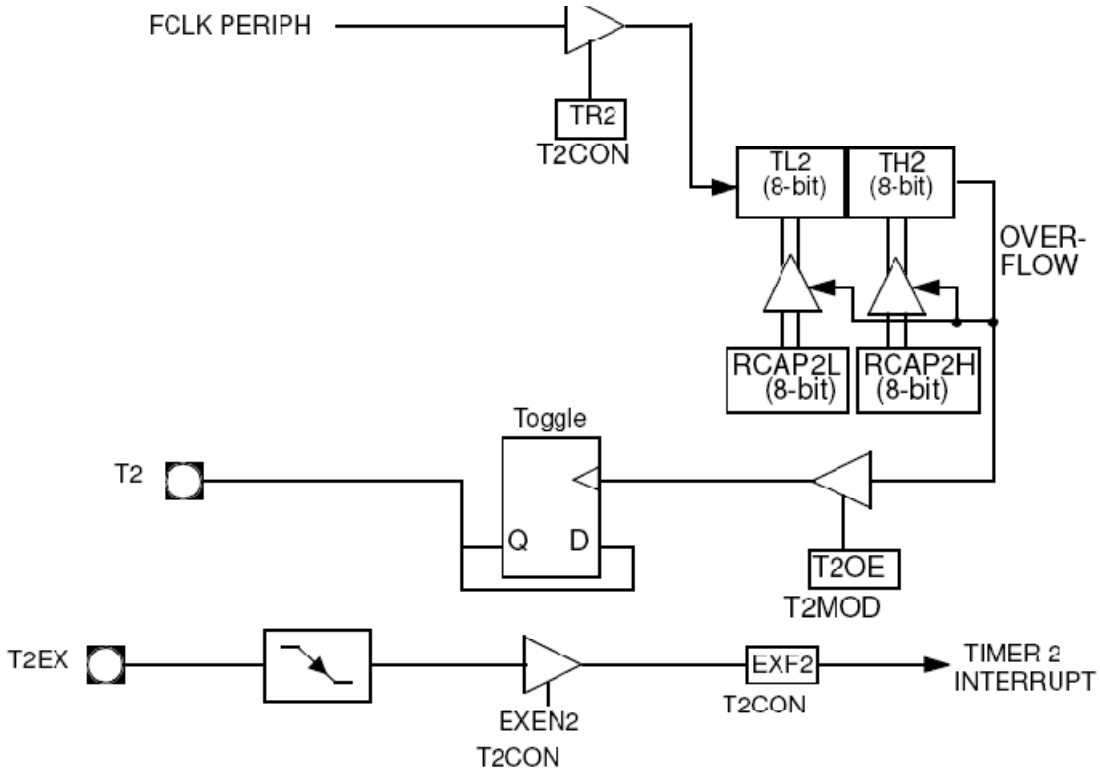
Trong chế độ này, timer tạo ra một xung clock có chu kỳ bổn phận (duty cycle) 50%. Khi timer tràn, nội dung của thanh ghi RCAP2H, RCAP2L được nạp vào cặp thanh ghi TH2, TL2 và timer tiếp tục đếm. Tần số xung clock tại chân T2 được xác định theo công thức sau:

$$f = \frac{f_{OSC} x 2^{X2}}{2\left(65536 - \frac{RCAP2H}{RCAP2L}\right)}$$

X2: bit nằm trong thanh ghi CKCON. Trong chế độ X2: $f_{OSC} = f_{thạch\ anh}$, ngược lại thì $f_{OSC} = f_{thạch\ anh}/2$.

Để timer 2 hoạt động ở chế độ tạo xung clock, cần thực hiện các bước sau:

- Đặt bit T2OE trong thanh ghi T2MOD = 1.
- Xoá bit $C/\overline{T}2$ trong thanh ghi T2CON = 0 (do chế độ này không cho phép đếm bằng dao động ngoài mà chỉ đếm bằng dao động nội).
- Xác định giá trị của cặp thanh ghi RCAP2H và RCAP2L theo tần số xung clock cần tạo.
- Khởi động giá trị cho cặp thanh ghi TH2, TL2 (có thể không cần thiết tuỳ theo ứng dụng).
- Đặt bit TR2 trong thanh ghi T2CON = 1 để cho phép timer chạy.



Hình 3.7 – Chế độ tạo xung clock

1.5.5. Chế độ tạo tốc độ baud

Khi các bit TCLK và RCLK trong thanh ghi T2CON được đặt lên mức 1, timer 2 sẽ dùng để tạo tốc độ baud cho cổng nối tiếp. Chế độ này cùng hoạt động như timer 0 và timer 1 (sẽ khảo sát cụ thể tại phần 2 – cổng nối tiếp).

1.6. Các ví dụ

Để điều khiển hoạt động của timer, cần thực hiện:

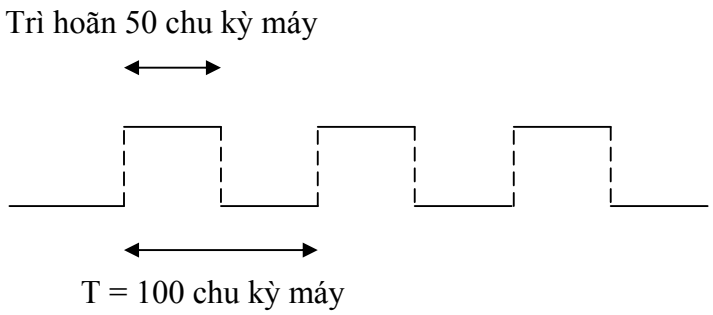
- Nạp giá trị cho thanh ghi TMOD để xác định chế độ hoạt động (thông thường chỉ dùng chế độ 1 – 16 bit và chế độ 2 – 8 bit tự động nạp lại).
- Nạp giá trị đếm trong các thanh ghi THx, TLx (thông thường sử dụng timer 0 và timer 1 nên quá trình đếm là đếm lên).
- Đặt các bit TR0, TR1 = 1 (cho phép timer hoạt động) hay xoá các bit này về 0 (cấm timer).
- Trong quá trình timer chạy, thực hiện kiểm tra các bit TF0, TF1 để xác định timer đã tràn hay chưa.
- Sau khi timer tràn, nếu thực hiện kiểm tra tràn bằng phần mềm (không dùng ngắt) thì phải thực hiện xoá TF0 hay TF1 để có thể tiếp tục hoạt động.

Ví dụ 1: Viết chương trình tạo sóng vuông tần số 10 KHz tại chân P1.0 dùng timer 0 (tần số thạch anh là f_{OSC} = 12MHz).

<u>Giải</u>

Do f_{OSC} = 12MHz nên chu kỳ máy = 1 µs.

f = 10 KHz → T = 1/f = 0.1 ms = 100 µs → một chu kỳ sóng vuông chiếm khoảng thời gian 100 chu kỳ máy → thời gian trì hoãn cần thiết là 50 chu kỳ máy.



Do giá trị đếm là 50 (ứng với 50 chu kỳ máy) nên chỉ cần dùng chế độ 8 bit (có thể đếm từ 1 đến 256) cho timer 0 (chế độ 2).

- Nội dung thanh ghi TMOD:

GATE1	C/T1	M11	M10	GATE0	C/T0	M01	M00
0	0	0	0	0	0	1	0
Timer 1 không dùng				Không dùng INT0	Đếm bằng dao động nội	Chế độ 8 bit	

TMOD = 0000 0010b (02h)

- Giá trị đếm là 50 và do timer 0 đếm lên nên giá trị cần nạp cho TH0 là -50 (có thể không cần nạp cho TL0 nhưng lúc đó chu kỳ đầu tiên của xung sẽ sai).

Chương trình thực hiện như sau:

```
     MOV TMOD,#02h
     MOV TH0,#(-50)
     MOV TL0,#(-50)
     SETB TR0       ; Cho phép timer 0 chạy
Lap:
     JNB TF0,Lap    ; Nếu Timer chưa tràn thì chờ
     CLR TF0
     CPL P1.0       ; Đảo bit P1.0 để tạo xung vuông
     SJMP Lap
     END
```

Ví dụ 2: Viết chương trình tạo xung vuông tần số f = 1 KHz tại P1.1 dùng timer 1(tần số thạch anh là f_{OSC} = 12MHz).

<u>Giải</u>

Do f_{OSC} = 12MHz nên chu kỳ máy = 1 µs.

f = 1 KHz → T = 1/f = 1 ms = 1000 µs → một chu kỳ sóng vuông chiếm khoảng thời gian 1000 chu kỳ máy → thời gian trì hoãn cần thiết là 500 chu kỳ máy.

Giá trị đếm là 500 vượt quá phạm vi của chế độ 8 bit nên phải sử dụng timer 1 ở chế độ 16 bit (chế độ 1). Đối với chế độ 16 bit, do không có giá trị nạp lại nên mỗi khi timer tràn, cần phải nạp lại giá trị cho thanh ghi TH1 và TL1.

- Nội dung thanh ghi TMOD:

GATE1	C/T1	M11	M10	GATE0	C/T0	M01	M00
0	0	0	1	0	0	0	0
Không dùng INT1	Đếm bằng dao động nội	Chế độ 16 bit		Timer 0 không dùng			

TMOD = 0001 0000b (10h)

- Giá trị đếm là 500 nên giá trị cần nạp cho cặp thanh ghi TH0_TL0 là -500 (dùng các lệnh giả HIGH và LOW).

Chương trình thực hiện như sau:

```
     MOV TMOD,#10h
Batdau:
     MOV TH1,#HIGH(-500)
     MOV TL1,#LOW(-500)
     SETB TR1       ; Cho phép timer 1 chạy
```

```
Lap:
     JNB TF1,Lap   ; Nếu Timer chưa tràn thì chờ
     CLR TF1
     CPL P1.1      ; Đảo bit P1.1 để tạo xung vuông
     CLR TR1
     SJMP Batdau   ; Quay lại nạp giá trị cho TH0_TL0
     END
```

Ví dụ 3: Viết chương trình tạo xung vuông tần số f = 10KHz tại P1.0 dùng timer 0 và xung vuông tần số f = 1 KHz tại P1.1 dùng timer 1.

Giải

Phân tích cho các thanh ghi giống như phần ví dụ 1 và 2 nhưng lưu ý rằng quá trình kiểm tra timer tràn sẽ khác: thực hiện kiểm tra timer 0, nếu chưa tràn thì kiểm tra timer 1 và kiểm tra tương tự cho timer 1.

Chương trình thực hiện như sau:

```
     MOV TMOD,#12h
     MOV TH1,#HIGH(-500)
     MOV TL1,#LOW(-500)
     MOV TH0,#(-50)
     MOV TL0,#(-50)
     SETB TR0
     SETB TR1
KtrT0:
     JNB TF0,KtrT1
     CLR TF0
     CPL P1.0
KtrT1:
     JNB  TF1,KtrT0
     CLR TF1
     CPL P1.1
     MOV TH1,#HIGH(-500)
     MOV TL1,#LOW(-500)
     SJMP KtrT0
     END
```

Lưu ý rằng, xung vuông tạo bằng cách như trên có thể không chính xác khi 2 timer tràn cùng lúc.

Ví dụ 4: Viết chương trình tạo xung vuông tần số f = 1 Hz tại P1.2 dùng timer1.

Giải

f = 1 Hz → T = 1/f = 1 s = 1 000 000 µs → một chu kỳ sóng vuông chiếm khoảng thời gian 500 000 chu kỳ máy → thời gian trì hoãn cần thiết là 500 000 chu kỳ máy.

Giá trị đếm là 500 000, vượt quá khả năng của timer (tối đa chỉ đếm được 65536 chu kỳ) nên phải thực hiện tạo vòng lặp đếm nhiều lần cho đến khi đạt đến giá trị 500 000 (có thể đếm mỗi lần 50 000 và thực hiện vòng lặp 10 lần).

Chương trình thực hiện như sau:

```
     MOV TMOD,#10h
Batdau:
     MOV R7,#10    ; Lặp 10 lần
Lap:
     MOV TH1,#HIGH(-50000)
     MOV TL1,#LOW(-50000)
     SETB TR1
KtrT1:
     JNB  TF1,KtrT1
     CLR TF1
     CLR  TR0
     DJNZ R7,Lap   ; Nếu R7 ≠ 0 thì lặp lại

     CPL P1.2      ; Đảo bit để tạo xung

     SJMP Batdau
     END
```

Ví dụ 5: Viết chương trình con tạo thời gian trì hoãn 1s dùng timer 0.

<u>Giải</u>

Do chương trình yêu cầu tạo thời gian trì hoãn nên số chu kỳ đếm là 1 000 000.

Chương trình như sau:

```
     MOV TMOD,#01h
     ;--- CHƯƠNG TRÌNH CHÍNH
     ;---

Delay1s:
     MOV R7,#20    ; Lặp 20 lần
Lap:
     MOV TH0,#HIGH(-50000)  ; Mỗi lần trì hoãn 50 000 µs
     MOV TL0,#LOW(-50000)
     SETB TR0
Lap1:
     JNB TF0,Lap1
     CLR TF0
     CLR TR0
     DJNZ R7,Lap   ; Lặp đủ 20 lần thì thoát
     RET
```

Lưu ý rằng khi viết chương trình trì hoãn như trên thì chương trình của AT89C51 xem như dừng lại, không làm gì cả (có thể giải quyết bằng cách sử dụng ngắt – xem thêm phần 3).

2. Cổng nối tiếp (Serial port)

Cổng nối tiếp trong 89C51 có khả năng hoạt động ở chế độ đồng bộ và bất đồng bộ dùng 2 chân TxD (P3.1) và RxD (P3.0). Chức năng của port nối tiếp là thực hiện chuyển đổi song song sang nối tiếp đối với dữ liệu xuất, và chuyển đổi nối tiếp sang song song đối với dữ liệu nhập.

Khi hoạt động ở chế độ truyền / nhận bất đồng bộ (UART – Universal Asynchronous Receiver / Transmitter), cổng nối tiếp có 3 chế độ song công (1, 2 và 3). Quá trình đọc / ghi cổng nối tiếp dùng thanh ghi SBUF (Serial Buffer), thực chất là 2 thanh ghi khác nhau: một thanh ghi truyền và một thanh ghi nhận.

Cổng nối tiếp có tất cả 4 chế độ khác nhau:

Chế độ 0: dữ liệu truyền / nhận thông qua chân RxD và xung clock dịch bit thông qua TxD với tốc độ baud bằng $f_{\text{thạch anh}}/12$.

Chế độ 1: truyền / nhận 10 bit: 1 bit start (luôn = 1), 8 bit dữ liệu và 1 bit stop (luôn = 0), tốc độ baud có thê thay đổi được và khi nhận, bit stop đưa vào RB8 của thanh ghi SCON.

Chế độ 2: truyền / nhận 11 bit: 1 bit start, 8 bit dữ liệu, bit thứ 9 và 1 bit stop. Khi truyền, bit 9 là bit TB8 và khi nhận, bit 9 là bit RB8 trong thanh ghi SCON. Tốc độ baud cố định là 1/32 hay 1/64 tần số thạch anh.

Chế độ 3: giống chế độ 2 nhưng tốc độ baud có thể thay đổi được.

Trong 4 chế độ trên, thường sử dụng chế độ 1 hay 3 để truyền dữ liệu. Trong trường hợp truyền dữ liệu giữa các vi điều khiển AT89C51 với nhau, có thể dùng chế độ 2. Ngoài ra, cổng nối tiếp còn có các chế độ nâng cao: kiểm tra lỗi khung và nhận dạng địa chỉ tự động.

2.1. Các thanh ghi điều khiển hoạt động

2.1.1. Thanh ghi SCON (Serial port controller)

Bảng 3.7 – Nội dung thanh ghi SCON

FE/SM0	SM1	SM2	REN	TB8	RB8	TI	RI

<table>
<tr><th>Bit</th><th>Ký hiệu</th><th>Địa chỉ</th><th>Mô tả</th></tr>
<tr><td rowspan="2">SCON.7</td><td>FE</td><td rowspan="2">9Fh</td><td>Framing Error – kiểm tra lỗi khung
Được đặt lên 1 khi phát hiện lỗi tại bit stop và phải xoá bằng phần mềm. Bit FE chỉ truy xuất được khi bit SMOD0 = 1 (trong thanh ghi PCON).</td></tr>
<tr><td>SM0</td><td>Serial port Mode bit 0 - Xác định chế độ cho cổng nối tiếp</td></tr>
<tr><td>SCON.6</td><td>SM1</td><td>9Eh</td><td>Serial port Mode bit 1
<table>
<tr><th>SM0</th><th>SM1</th><th>Mô tả</th><th>Tốc độ baud</th></tr>
<tr><td>0</td><td>0</td><td>Thanh ghi dịch</td><td>$f_{OSC}/12$</td></tr>
<tr><td>0</td><td>1</td><td>UART 8 bit</td><td>Thay đổi</td></tr>
<tr><td>1</td><td>0</td><td>UART 9 bit</td><td>$f_{OSC}/32$ hay $f_{OSC}/64$</td></tr>
<tr><td>1</td><td>1</td><td>UART 9 bit</td><td>Thay đổi</td></tr>
</table></td></tr>
<tr><td>SCON.5</td><td>SM2</td><td>9Dh</td><td>Serial port Mode bit 2 – Chế độ đa xử lý
= 0: bình thường
= 1: cho phép truyền thông đa xử lý trong chế độ 2 và 3</td></tr>
<tr><td>SCON.4</td><td>REN</td><td>9Ch</td><td>Reception Enable bit – Cho phép thu
= 0: cấm thu
= 1: cho phép thu tại cổng nối tiếp</td></tr>
<tr><td>SCON.3</td><td>TB8</td><td>9Bh</td><td>Transmitter Bit – Bit truyền thứ 9 trong chế độ 2 và 3</td></tr>
<tr><td>SCON.2</td><td>RB8</td><td>9Ah</td><td>Receiver Bit – Bit nhận thứ 9 trong chế độ 2 và 3. Trong chế độ 1, nếu SM2 = 0 thì RB8 = stop bit.</td></tr>
<tr><td>SCON.1</td><td>TI</td><td>99h</td><td>Transmit Interrupt flag – Cờ ngắt phát
Được đặt bằng 1 khi kết thúc quá trình truyền và xoá bằng phần mềm.</td></tr>
<tr><td>SCON.0</td><td>RI</td><td>99h</td><td>Receive Interrupt flag – Cờ ngắt thu
Được đặt bằng 1 khi nhận xong dữ liệu và xoá bằng phần mềm.</td></tr>
</table>

Giá trị khi reset: 00h, cho phép định địa chỉ bit

2.1.2. Thanh ghi BDRCON (Baud Rate Control Register)

Bảng 3.8 – Nội dung thanh ghi BDRCON

-	-	-	BRR	TBCK	RBCK	SPD	SRC

Bit	Ký hiệu	Mô tả
7	-	
6	-	
5	-	
4	BRR	Baud Rate Run control bit – Cho phép hoạt động = 0: cấm bộ tạo tốc độ baud nội (internal baud rate generator) hoạt động = 1: cho phép
3	TBCK	Transmission Baud rate generator selection bit for UART – Chọn bộ tạo tốc độ baud truyền là bộ tạo tốc độ nội (= 1) hay bằng timer (= 0)
2	RBCK	Reception Baud rate generator selection bit for UART – Chọn bộ tạo tốc độ baud nhận là bộ tạo tốc độ nội (= 1) hay bằng timer (= 0)
1	SPD	Baud Rate Speed control bit for UART – Chọn tốc độ baud là nhanh (= 1) hay chậm (= 0)
0	SRC	Baud Rate Source select bit in Mode 0 for UART – Chọn tốc độ baud trong chế độ 0 từ dao động thạch anh (= 0) hay từ bộ tạo tốc độ baud nội (= 1)

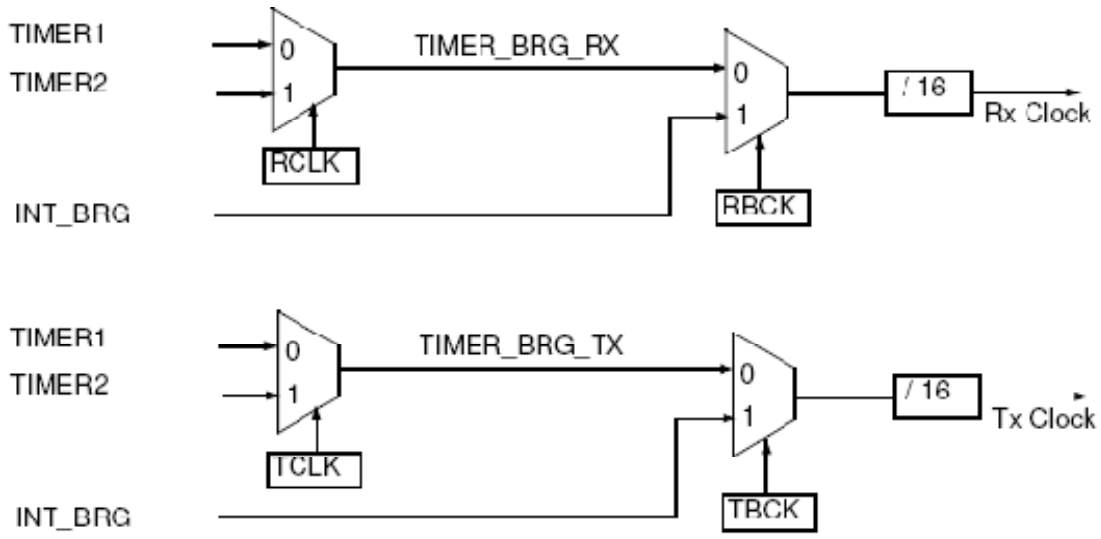
Giá trị khi reset: 00h, không cho phép định địa chỉ bit

Ngoài ra còn có các thanh ghi SBUF (Serial Buffer), BRL (Baud Rate Reload), SADEN (Slave Address Mark), SADDR (Slave Address).

Lưu ý rằng các thanh ghi BDRCON, BRL, SADEN và SADDR chỉ có trong các phiên bản mới của MCS-51.

2.2. Tạo tốc độ baud

- Chế độ 0: tốc độ baud cố định = 1/12 tần số thạch anh.
- Chế độ 2: tốc độ baud = 1/32 tần số thạch anh khi SMOD = 1 hay 1/64 khi SMOD = 0 (SMOD: nằm trong thanh ghi PCON).
- Chế độ 1 và 3: tốc độ baud xác định bằng tốc độ tràn của timer 1. Trong họ 89x52, có thể dùng timer 2 để tạo tốc độ baud còn trong các phiên bản mới, có thể dùng bộ tạo tốc độ nội (INT_BRG – Internal Baud Rate Generator). Việc xác định nguồn tạo tốc độ baud mô tả như hình 3.8 và bảng 3.9.



Hình 3.8 – Lựa chọn tốc độ baud

Bảng 3.9 – Lựa chọn tốc độ baud

TCLK	RCLK	TBCK	RBCK	Clock phát	Clock thu
0	0	0	0	Timer 1	Timer 1
1	0	0	0	Timer 2	Timer 1
0	1	0	0	Timer 1	Timer 2
1	1	0	0	Timer 2	Timer 2
X	0	1	0	INT_BRG	Timer 1
X	1	1	0	INT_BRG	Timer 2
0	X	0	1	Timer 1	INT_BRG
1	X	0	1	Timer 2	INT_BRG
X	X	1	1	INT_BRG	INT_BRG

2.2.1. Tạo tốc độ baud bằng Timer 1

Khi dùng timer 1 để tạo tốc độ baud, thông thường cần thiết lập timer 1 hoạt động ở chế độ 8 bit tự nạp lại và giá trị nạp ban đầu của timer 1 (chứa trong thanh ghi TH1) phụ thuộc vào tốc độ baud cần tạo theo công thức sau:

$$\text{Giá trị nạp} = -\frac{f_{OSC} \times 2^{SMOD}}{12 \times 32 \times \text{baud_rate}}$$

Ví dụ: Giả sử tần số thạch anh là f_{OSC} = 11.0592 MHz, giá trị nạp khi tạo tốc độ baud 4800 bps là:

Nếu SMOD = 0: giá trị nạp = $-\frac{11.0592\times10^6\times2^0}{12\times32\times4800}=-6\rightarrow$ TH1 = -6 hay TH1 = FAh

Nếu SMOD = 1: giá trị nạp = $-\frac{11.0592\times10^6\times2^1}{12\times32\times4800}=-12\rightarrow$ TH1 = -12 hay TH1 = F4h

Ví dụ: Giả sử tần số thạch anh là f_{OSC} = 12 MHz, giá trị nạp khi tạo tốc độ baud 4800 bps là:

Nếu SMOD = 0: giá trị nạp = $-\frac{12\times10^6\times2^0}{12\times32\times4800}=-6.51\rightarrow$ chọn giá trị nạp là -6 hay -7. Nếu chọn giá trị nạp = -6 thì tốc độ baud = 5208 bps còn nếu chọn -7 thì tốc độ baud là 4464 bps.

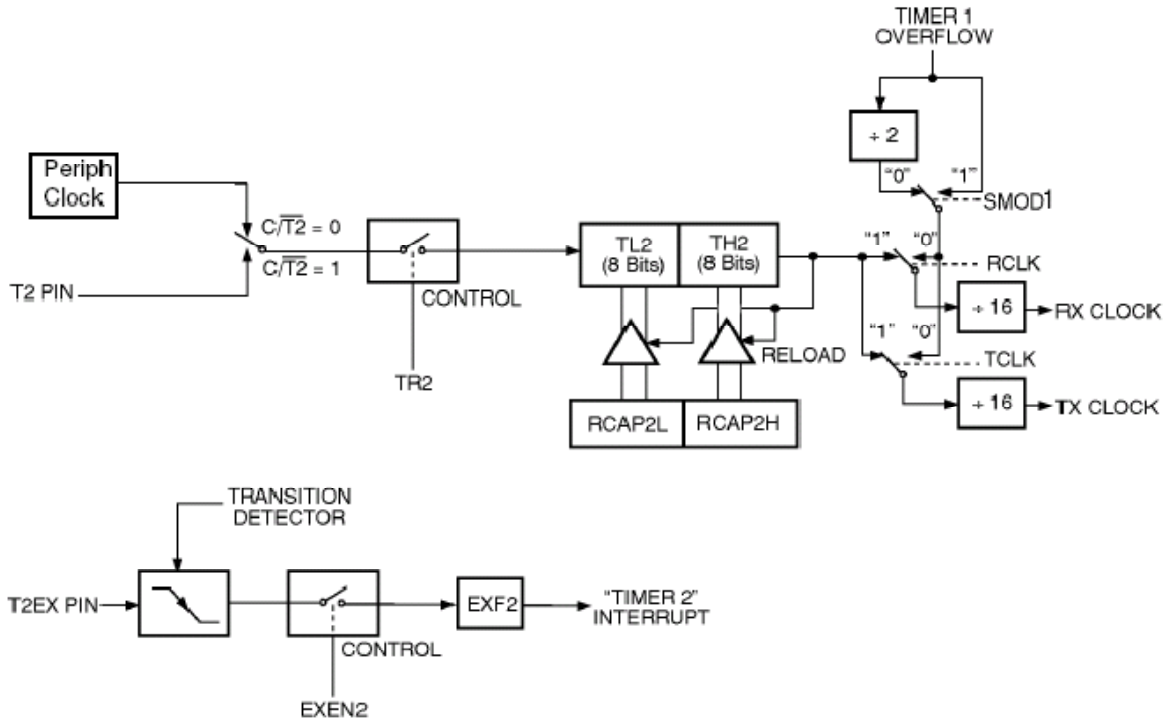
Nếu SMOD = 1: giá trị nạp = $-\frac{11.0592\times10^6\times2^1}{12\times32\times4800}=-13.02\rightarrow$ chọn giá trị nạp là -13 $\rightarrow$ tốc độ baud là 4807 bps. Như vậy, khi dùng tần số thạch anh là 12 MHz thì tốc độ baud sẽ có sai số $\rightarrow$ chỉ dùng khi kết nối nhiều vi điều khiển MCS-51 với nhau còn khi kết nối với các thiết bị khác (như máy tính chẳng hạn) thì nên sử dụng tần số thạch anh 11.0592 MHz.

Các giá trị nạp thông dụng cho MCS-51 mô tả như sau:

Bảng 3.10 – Các giá trị nạp thông dụng

Tốc độ [bps]	f_{OSC}[MHz]	SMOD	Giá trị nạp	Tốc độ thực [bps]	Sai số
1200	11.059	0	-12	1200	0
4800	11.059	0	-6	4800	0
9600	11.059	0	-3	9600	0
1200	11.059	1	-24	1200	0
19200	11.059	1	-3	19200	0
1200	12	0	-26	1201.9	2.17%
2400	12	0	-13	2403.8	0.16%
4800	12	0	-6	5208.3	8.5%
9600	12	0	-3	10416.7	8.5%

2.2.2. Tạo tốc độ baud bằng Timer 2



Hình 3.9 – Tạo tốc độ baud bằng timer 2

Timer 2 được dùng để tạo tốc độ baud khi đặt các bit TCLK, RCLK lên 1 (trong thanh ghi T2CON). Công thức liên quan giữa tốc độ baud và giá trị nạp như sau (lưu ý rằng giá trị nạp chứa trong cặp thanh ghi RCAP2H_RCAP2L):

$$\text{Giá trị nạp} = -\frac{f_{OSC}}{2 \times 16 \times \text{baud_rate}}$$

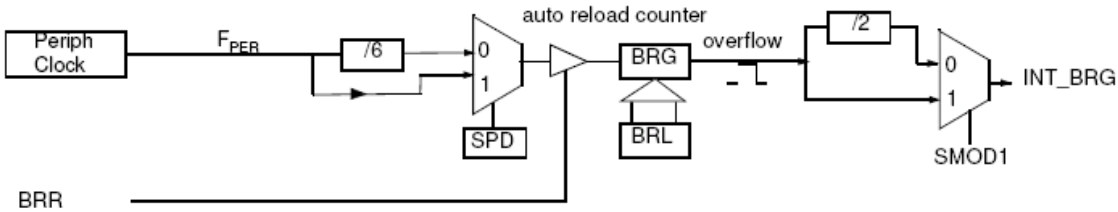
Khi dùng Timer 2 để tạo tốc độ baud, xung clock thu và phát có thể tách riêng bằng cách chỉ dùng TCLK hay RCLK. Lúc đó, xung clock còn lại được xác định theo Timer 1. Ngoài ra, cũng có thể tạo ngắt cho Timer 2 bằng cách đặt bit EXEN2 = 1 và ngắt tạo ra khi xuất hiện cạnh âm tại chân T2EX.

Ví dụ: Giả sử tần số thạch anh là f_{OSC} = 11.0592 MHz, giá trị nạp khi tạo tốc độ baud 4800 bps là:

$$\text{Giá trị nạp} = -\frac{11.0592 \times 10^6}{2 \times 16 \times 4800} = -72 \rightarrow \text{FFB8h}$$

$\rightarrow$ RCAP2H = FFh, RCAP2L = B8h

2.2.3. Bộ tạo tốc độ baud nội (INT_BRG – Internal Baud Rate Generator)



Hình 3.10 – Bộ tạo tốc độ baud nội

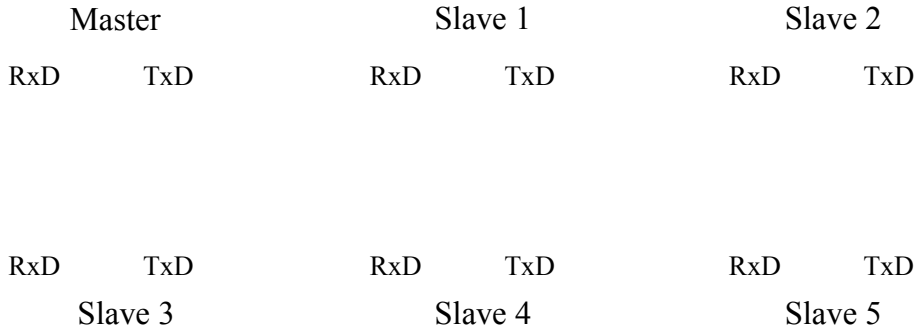
Giá trị nạp trong bộ tạo tốc độ nội chứa trong thanh ghi BRL và được xác định theo công thức sau:

$$\text{Giá trị nạp} = -\frac{f_{OSC} \times 2^{SMOD1}}{2 \times 32 \times 6^{1-SPD} \times baud_rate}$$

Trong đó SMOD1 nằm trong thanh ghi PCON và SPD nằm trong thanh ghi BDRCON.

2.3. Truyền thông đa xử lý

Chế độ 2 và 3 của MCS-51 cho phép thực hiện kết nối nhiều vi điều khiển ở chế độ master – slave. Mô hình thực hiện của quá trình truyền thông mô tả như hình vẽ sau:



Hình 3.11 – Truyền thông đa xử lý

Quá trình truyền dữ liệu mô tả như sau:

- Khi khởi động, các vi điều khiển slave có bit SM2 = 1 (trong thanh ghi SCON) và hoạt động ở chế độ UART 9 bit. Như vậy, slave chỉ nhận được dữ liệu khi bit truyền thứ 9 (TB8 của master) là 1.

- Mỗi slave được gán trước một địa chỉ. Khi cần trao đổi thông tin với slave nào, master sẽ gởi dữ liệu 9 bit gồm 8 bit địa chỉ của slave và bit 9 = 1. Dữ liệu này sẽ được tất cả các slave nhận về (do bit 9 = 1). Chương trình trong slave sẽ kiểm tra giá trị địa chỉ tương ứng, nếu trùng với địa chỉ đã cài đặt sẵn thì đảo bit SM2 (= 0), nếu khác thì bỏ qua.

- Tiếp tục, master sẽ gởi dữ liệu đến slave nhưng lúc này bit 9 = 0. Khi đó, chỉ có slave nào có bit SM2 = 0 mới nhận được dữ liệu.

- Sau khi truyền xong dữ liệu, master gởi lại 8 bit địa chỉ và bit 9 = 1. Slave nhận được sẽ đảo bit SM2 lần nữa để khôi phục trạng thái ban đầu.

Như vậy, trong quá trình truyền thông đa xử lý, có 2 loại thông tin gởi: byte địa chỉ nếu bit 9 = 1 và byte dữ liệu nếu bit 9 = 0.

2.4. Nhận dạng địa chỉ tự động

Trong các phiên bản mới của MCS-51, địa chỉ của các slave có thể nhận dạng bằng các thanh ghi SADDR và thanh ghi mặt nạ SADEN (các bit không quan tâm trong thanh ghi địa chỉ SADDR sẽ tương ứng với các bit 0 trong thanh ghi SADEN).

Xét hệ thống có 1 master và 3 slave:

Slave 1: SADDR = 1111 0001b, SADEN = 1111 1010b

```
 1111 0001b
 1111 1010b
 ----------
 1111 0x0xb
```

Slave 2: SADDR = 1111 0011b, SADEN = 1111 1001b

```
 1111 0011b
 1111 1001b
 ----------
 1111 0xx1b
```

Slave 3: SADDR = 1111 0001b, SADEN = 1111 1010b

```
 1111 1011b
 1111 0101b
 ----------
 1111 x0x1b
```

Nếu chỉ cần gởi dữ liệu cho slave 1, địa chỉ cần sử dụng có bit 0 = 0 (do địa chỉ của slave 2 và slave 3 có bit 0 = 1 còn địa chỉ của slave 1 có bit 0 tuỳ ý), giả sử là 1111 0000b.

Nếu cần gởi cho slave 2 và slave 3 mà không gởi cho slave 1 thì địa chỉ cần dùng có bit 1 = 1 (do địa chỉ của slave 1 có bit 1 = 0 còn slave 2 và 3 thì tuỳ ý), giả sử như 1111 0011b.

❖ **Địa chỉ broadcast**

Địa chỉ broadcast tạo thành từ phép toán OR giữa các thanh ghi SADDR và SADEN trong đó các bit 0 xác định đó là các bit không quan tâm.

Giả sử SADDR = 0101 0000b và SADEN = 1111 1101b thì

```
      0101 0000b
OR    1111 1101b
    ------------
      1111 1101b
```

Địa chỉ broadcast là 1111 11x1b.

2.5. Kiểm tra lỗi khung

Chế độ kiểm tra lỗi khung chỉ có trong các chế độ 1, 2 và 3 được thực hiện bằng cách đặt bit SMOD0 lên 1 (trong thanh ghi PCON). Khi SMOD0 = 1, bộ thu sẽ kiểm tra bit stop mỗi khi có dữ liệu đến. Nếu bit stop không hợp lệ, bit FE sẽ được đặt lên 1 (trong thanh ghi SCON).

Phần mềm sau khi đọc byte dữ liệu sẽ kiểm tra bit FE để xác định có lỗi đường truyền hay không. Lưu ý rằng bit FE chỉ xoá bằng phần mềm hay khi reset hệ thống mà không bị xoá khi nhận bit stop hợp lệ.

2.6. Các ví dụ

Để điều khiển hoạt động của cổng nối tiếp, cần thực hiện các bước sau:

- Khởi động giá trị của thanh ghi SCON để xác định chế độ hoạt động.
- Chọn bộ tạo tốc độ baud (mặc định là timer 1) và xác định các thông số cần thiết theo tốc độ baud yêu cầu.
- Kiểm tra các bit TI và RI để xác định cho phép truyền hay nhận dữ liệu không.
- Nếu cần truyền dữ liệu thì kiểm tra TI và chuyển nội dung truyền vào thanh ghi SBUF.

- Nếu cần nhận dữ liệu thì kiểm tra RI và đọc nội dung từ SBUF vào thanh ghi A.

Ví dụ 1: Khởi động cổng nối tiếp ở chế độ UART 8 bit với tốc độ baud 9600 bps, dùng timer 1 là bộ tạo tốc độ baud (giả sử tần số thạch anh là 11.0592 MHz).

Giải

- Nội dung thanh ghi SCON:

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
0	1	0	1	0	0	1	0
UART 8 bit		Không ở chế độ đa xử lý	Cho phép thu			Cho phép truyền	

SCON = 0101 0010b (52h)

- Nội dung thanh ghi TMOD:

GATE1	C/T1	M11	M10	GATE0	C/T0	M01	M00
0	0	1	0	0	0	0	0
Không dùng INT1	Đếm bằng dao động nội	Chế độ 8 bit		Timer 0 không dùng			

TMOD = 0010 0000b (20h)

- Giá trị đếm (theo bảng 3.10): TH1 = -3

Đoạn chương trình khởi động như sau:

```
MOV SCON,#52h
MOV TMOD, #20h
MOV TH1,#-3
SETB TR1
```

Ví dụ 2: Viết chương trình xuất liên tục các ký tự từ ‘A’ đến ‘Z’ ra cổng nối tiếp với tốc độ baud 4800 bps (giả sử tần số thạch anh là 11.0592 MHz).

Giải

Tốc độ = 4800 bps → giá trị đếm: TH1 = -6

Chương trình thực hiện như sau:

```
     MOV SCON,#52h
     MOV TMOD,#20h
     MOV TH1,#-6
     SETB TR1
Batdau:
     MOV A,#'A'
```

```
Truyen:
     JNB TI,$ ; Nếu chưa cho phép truyền thì chờ
     CLR TI   ; Xoá TI để không cho phép truyền, sau khi
              ; truyền xong thì mới có thể truyền tiếp
     MOV SBUF,A    ; Truyền dữ liệu
     INC A         ; Qua ký tự kế
     CJNE A,#'Z'+1,Truyen; Nếu đã truyền xong từ 'A'
     SJMP Batdau        ; đến 'Z' thì lặp lại quá trình
```

Ví dụ 3: Viết chương trình nhận ký tự từ cổng nối tiếp với tốc độ baud 19200bps (giả sử tần số thạch anh là 11.0592 MHz).

Giải

Tốc độ = 1900 bps → giá trị đếm: TH1 = -3 và SMOD = 1

Chương trình thực hiện như sau:

```
     MOV SCON,#52h
     MOV TMOD,#20h
     MOV A,PCON    ; Gán bit SMOD = 1 (do PCON không cho
     SETB ACC.7    ; phép định địa chỉ bit nên phải thực
     MOV PCON,A    ; hiện thông qua thanh ghi A)

     MOV TH1,#-3
     SETB TR1
Nhan:
     JNB RI,$ ; Nếu chưa có ký tự đến thì chờ
     CLR RI   ; Xoá RI để không cho phép nhận, sau khi
              ; có ký tự tiếp theo thì mới nhận
     MOV A,SBUF    ; Nhận dữ liệu
     SJMP Nhan
```

Lưu ý rằng, đối với các ví dụ trên, khi truyền hay nhận dữ liệu thì MCS-51 phải chờ, không được thực hiện công việc khác. Vấn đề này có thể giải quyết bằng cách sử dụng ngắt (xem thêm phần 3).

3. Ngắt (Interrupt)

Ngắt là quá trình dừng chương trình đang thực thi để phục vụ cho một chương trình khác khi xảy ra một sự kiện. Chương trình xử lý sự kiện ngắt gọi là chương trình phục vụ ngắt (ISR – Interrupt Service Routine).

Họ MCS-51 có tổng cộng 5 nguồn ngắt khác nhau (không kể reset cũng có thể xem như là một ngắt): ngắt ngoài 0, 1 (tại các chân $\overline{INT0}$, $\overline{INT1}$), timer 0, 1 (khi timer tương ứng tràn), cổng nối tiếp (khi có ký tự đến hay khi truyền ký tự đi). Đối với họ 89x52 sẽ có thêm ngắt timer 2.

3.1. Các thanh ghi điều khiển hoạt động

3.1.1. Thanh ghi IE (Interrupt Enable)

Bảng 3.11 – Nội dung thanh ghi IE

EA	-	ET2	ES	ET1	EX1	ET0	EX0

Bit	Ký hiệu	Địa chỉ	Mô tả
IE.7	EA	AFh	Enable All Cấm tất cả (= 0) hay cho phép ngắt
IE.6	-		
IE.5	ET2	ADh	Enable Timer 2 Cho phép ngắt tại timer 2 (= 1)
IE.4	ES	ACh	Enable serial port Cho phép ngắt tại cổng nối tiếp (= 1)
IE.3	ET1	ABh	Enable Timer 1 Cho phép ngắt tại timer 1 (= 1)
IE.2	EX1	AAh	Enable External interrupt 1 Cho phép ngắt tại ngắt ngoài 1 (= 1)
IE.1	ET0	A9h	Enable Timer 0 Cho phép ngắt tại timer 0 (= 1)
IE.0	EX0	A8h	Enable External interrupt 0 Cho phép ngắt tại ngắt ngoài 0 (= 1)

Giá trị khi reset: 00h, cho phép định địa chỉ bit

Thanh ghi IE cho phép một ngắt có xảy ra hay cấm ngắt (để cho phép cần dùng 2 bit: bit EA = 1 và bit cho phép tương ứng từng ngắt).

3.1.2. Thanh ghi IP (Interrupt Priority)

Bảng 3.12 – Nội dung thanh ghi IP

-	-	PT2	PS	PT1	PX1	PT0	PX0

Bit	Ký hiệu	Địa chỉ	Mô tả
IP.7	-		
IP.6	-		
IP.5	PT2	BDh	Chọn mức ưu tiên cao (= 1) hay thấp (= 0) tại timer 2
IP.4	PS	BCh	Chọn mức ưu tiên cao (= 1) hay thấp (= 0) tại cổng nối tiếp

IP.3	PT1	BBh	Chọn mức ưu tiên cao (= 1) hay thấp (= 0) tại timer 1
IP.2	PX1	BAh	Chọn mức ưu tiên cao (= 1) hay thấp (= 0) tại ngắt ngoài 1
IP.1	PT0	B9h	Chọn mức ưu tiên cao (= 1) hay thấp (= 0) tại timer 0
IP.0	PX0	B8h	Chọn mức ưu tiên cao (= 1) hay thấp (= 0) tại ngắt ngoài 0

Giá trị khi reset: 00h, cho phép định địa chỉ bit

Thanh ghi IP cho phép chọn mức ưu tiên cho các ngắt. Họ MCS-51 có 2 mức ưu tiên: mức cao và mức thấp. Quá trình xử lý ưu tiên ngắt mô tả như sau:

- Nếu 2 ngắt xảy ra đồng thời thì ngắt nào có mức ưu tiên cao hơn sẽ được phục vụ trước.
- Nếu 2 ngắt xảy ra đồng thời có cùng mức ưu tiên thì thứ tự ưu tiên thực hiện từ cao đến thấp như sau: ngắt ngoài 0 – timer 0 – ngắt ngoài 1 – timer 1 – cổng nối tiếp – timer 2.
- Nếu ISR của một ngắt có mức ưu tiên thấp đang chạy mà có ngắt khác xảy ra với mức ưu tiên cao thì ISR này sẽ tạm dừng để chạy ISR có mức ưu tiên cao (cũng có nghĩa là không thể dừng ISR có mức ưu tiên cao).

3.1.3. Thanh ghi TCON (Timer/Counter Control)

Bảng 3.13 – Nội dung thanh ghi TCON

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit	Ký hiệu	Địa chỉ	Mô tả
TCON.7	TF1	8Fh	Xem phần timer
TCON.6	TR1	8Eh	
TCON.5	TF0	8Dh	
TCON.4	TR0	8Ch	
TCON.3	IE1	8Bh	Cờ ngắt ngoài 1 Đặt bằng 1 khi phát hiện tác động ngắt tại $\overline{INT1}$ Xoá bằng phần mềm hay bằng phần cứng khi chuyển điều khiển đến ISR
TCON.2	IT1	8Ah	Interrupt 1 Type control bit = 0: ngắt ngoài 1 được tác động bằng mức logic 0 = 1: ngắt ngoài 1 được tác động bằng cạnh âm
TCON.1	IE0	89h	Dùng cho ngắt ngoài 0
TCON.0	IT0	88h	

Giá trị khi reset: TCON = 00h

3.2. Xử lý ngắt

Để kiểm tra khi nào ngắt xảy ra, các cờ ngắt được lấy mẫu ở thời gian S5P2 của mỗi chu kỳ máy. Các điều kiện ngắt được hỏi vòng cho đến chu kỳ máy kế tiếp để xác định xem có ngắt xảy ra hay không. Khi có điều kiện ngắt, hệ thống ngắt sẽ tạo ra lệnh LCALL để gọi ISR tương ứng nhưng lệnh này sẽ không được thực hiện khi tồn tại một trong các điều kiện sau:

- Có một ngắt có mức ưu tiên bằng hay cao hơn đang được phục vụ.
- Chu kỳ hỏi vòng hiện tại không phải là chu kỳ cuối của một lệnh.
- Đang thực thi lệnh RETI hay bất kỳ lệnh nào có ảnh hưởng đến thanh ghi IE và IP.

Khi có ngắt xảy ra, các thao tác thực hiện lần lượt là:

- Hoàn tất lệnh hiện hành.
- Cất nội dung của thanh ghi PC vào stack.
- Lưu trạng thái của ngắt hiện hành.
- Đưa vào thanh ghi PC địa chỉ của ISR tương ứng.

Sau khi thực hiện xong ISR (kết thúc bằng lệnh RETI), thực hiện quá trình: khôi phục trạng thái ban đầu của ngắt và lấy địa chỉ từ stack đưa vào PC.

❖ **Bảng vector ngắt**

Khi xảy ra ngắt, thanh ghi PC sẽ được nạp giá trị tương ứng với các ngắt. Các giá trị này được gọi là vector ngắt, mô tả như sau:

Bảng 3.14 – Bảng vector ngắt

Nguyên nhân ngắt	**Địa chỉ**
Reset	0000h
Ngắt ngoài 0	0003h
Timer 0	000Bh
Ngắt ngoài 1	0013h
Timer 1	001Bh
Cổng nối tiếp	0023h
Timer 2	002Bh

Trong các nguyên nhân này, reset có thể được xem như một ngắt có vector ngắt là 0000h nhưng cách xử lý khi reset không giống như ngắt: khởi động tất cả các thanh ghi về giá trị mặc định và không lưu nội dung của PC vào stack.

Theo bảng vector ngắt, ISR của ngắt ngoài 0 nằm từ địa chỉ 0003h đến 000Ah (chiếm tổng cộng 8 byte) nên khi sử dụng ISR có kích thước thấp hơn 9 byte thì có thể dùng trực tiếp tại địa chỉ 0003h (xem thêm phần sau). Tuy nhiên, nếu kích thước ISR lớn hơn thì phải dùng các lệnh nhảy tại các vector ngắt. Khi đó chương trình sẽ có cấu trúc như sau (tên của các ISR có thể thay đổi):

```
     ORG 0000h
     LJMP main
     ORG 0003h
     LJMP Int0_ISR
     ORG 000Bh
     LJMP Timer0_ISR
     ORG 0013h
     LJMP Int1_ISR
     ORG 001Bh
     LJMP Timer1_ISR
     ORG 0023h
     LJMP Serial_ISR
Main:
     ……
     ……
Int0_ISR:
     ……
     RETI
Timer0_ISR:
     ……
     RETI
Int1_ISR:
     ……
     RETI
Timer1_ISR:
     ……
     RETI
Serial_ISR:
     ……
     RETI
END
```

Lưu ý rằng nếu không sử dụng ngắt nào thì không cần phải khai báo ISR cho ngắt đó.

3.3. Ngắt do bộ định thời

MCS-51 có 2 nguồn ngắt từ timer: timer 0 và timer 1 (đối với họ 89x52 còn có thêm timer 2). Khi timer hoạt động ở chế độ ngắt, chương trình vẫn hoạt động bình thường cho đến khi timer tràn thì mới chuyển đến vị trí của ISR (trong khi đó, khi timer hoạt động không sử dụng ngắt thì chương trình sẽ dừng lại – xem thêm phần ví dụ trong hoạt động định thời).

Các nguồn ngắt này cho phép hay cấm bằng các bit trong thanh ghi IE: EA, ET0, ET1 và chọn chế độ ưu tiên bằng các bit trong thanh ghi IP: PT0, PT1. Khi timer tràn, cờ TFx sẽ chuyển lên mức 1. Hệ thống ngắt khi phát hiện cờ TFx lên 1 sẽ chuyển đến ISR tương ứng và tự động xoá cờ TFx.

Quá trình điều khiển hoạt động bằng bộ định thời có sử dụng ngắt thực hiện như sau:

- Xác định chế độ hoạt động của bộ định thời.
- Nạp giá trị cho các thanh ghi THx, TLx.
- Cho phép ngắt tại các bộ định thời tương ứng (thanh ghi IE).
- Xác định mức ưu tiên (thanh ghi IP).
- Cho phép timer chạy bằng các bit TRx.
- Viết ISR cho timer tương ứng.

Ví dụ 1: Viết chương trình tạo sóng vuông tần số f = 5 KHz tại P1.0 dùng ngắt timer 1 (giả sử tần số thạch anh là 12 MHz).

<u>Giải</u>

f = 5 KHz → T = 200 µs (200 chu kỳ) → thời gian trì hoãn: 100 chu kỳ

Giá trị đếm = 100 → dùng chế độ 8 bit

TMOD = 0010 0000b (20h)

- Nội dung thanh ghi IE:

EA	-	ET2	ES	ET1	EX1	ET0	EX0
1	0	0	0	1	0	0	0

IE = 1000 1000b (88h)

Chương trình thực hiện như sau:

```
ORG 0000h
```

```
     LJMP main
     ORG 001Bh
     CPL P1.0       ; đảo bit
     RETI           ; trở về chương trình chính từ ISR
Main:
     MOV TMOD,#20h
     MOV IE,#88h    ; Có thể thay thế bằng 2 lệnh sau:
                    ; SETB EA
                    ; SETB ET1
     MOV TH1,#(-100)
     MOV TL1,#(-100)
     SETB TR1
     SJMP $         ; Lặp tại chỗ, nghĩa là chương trình
                    ; không làm gì cả, chờ timer tràn (các
                    ; ứng dụng thực tế có thể xử lý các
                    ; công việc khác)
     END
```

Lưu ý rằng lệnh CPL P1.0 chiếm 2 byte, lệnh RETI chiểm 1 byte, tổng cộng ISR cho timer 1 là 3 byte không vượt quá 8 byte nên có thể đặt trực tiếp tại địa chỉ 001Bh.

Ví dụ 2: Viết chương trình tạo xung vuông tần số f = 10KHz tại P1.0 dùng ngắt timer 0 và xung vuông tần số f = 1 KHz tại P1.1 dùng ngắt timer 1.

<u>Giải</u>

Giá trị đếm cho timer 0: 50.

Giá trị đếm cho timer 1: 500.

→ timer 0: 8 bit, timer 1: 16 bit

TMOD = 0001 0010b (12h)

- Nội dung thanh ghi IE:

EA	-	ET2	ES	ET1	EX1	ET0	EX0
1	0	0	0	1	0	1	0

IE = 1000 1010b (8Ah)

Chương trình thực hiện như sau:

```
     ORG 0000h
     LJMP main
```

```
     ORG 000Bh
     CPL P1.0
     RETI
     ORG 001Bh
     MOV TH1,#HIGH(-500)     ; 2 byte
     MOV TL1,#LOW(-500)      ; 2 byte
     CPL P1.1                ; 2 byte
     RETI                    ; 1 byte
Main:
     MOV TMOD,#12h
     MOV IE,#8Ah
     SETB TR0
     SETB TR1
     MOV TH1,#HIGH(-500)
     MOV TL1,#LOW(-500)
     MOV TH0,#(-50)
     MOV TL0,#(-50)
     SJMP $
     END
```

Trong ví dụ này, do timer 1 hoạt động ở chế độ 16 bit nên mỗi lần timer tràn phải thực hiện nạp lại giá trị cho timer 1.

Ví dụ 3: Viết chương trình dùng ngắt timer 0 sao cho cứ 1s thì tăng nội dung của các ô nhớ 30h, 31h, 32h theo quy luật đồng hồ (30h chứa giờ, 31h chứa phút, 32h chứa giây).

Giải

Yêu cầu chương trình trì hoãn là 1s trong khi timer 0 cho phép trì hoãn tối đa là 65536 µs → chọn giá trị đếm là 50000 và thực hiện lặp lại 20 lần (20 x 50000 = 1000000 µs = 1s).

TMOD = 0000 0001b (01h)

IE = 1000 0010b (82h)

Chương trình thực hiện như sau:

```
     Hour EQU 30h        ; Định nghĩa trước các ô nhớ
     Minute EQU 31h
     Second EQU 32h
     ORG 0000h
     LJMP main
```

```
     ORG 0003h
     LJMP Timer0_ISR
Main:
     MOV TMOD,#01h
     MOV IE,#82h
     MOV TH0,#HIGH(-50000)
     MOV TL0,#LOW(-50000)
     MOV R7,#20              ; Lặp 20 lần
     SETB TR0
     SJMP $
Timer0_ISR:
     MOV TH0,#HIGH(-50000)
     MOV TL0,#LOW(-50000)
     DJNZ R7,exitTimer0      ; Nếu chưa đủ 20 lần thì thoát
     CALL Inc_clock          ; Tăng theo quy luật đồng hồ
     MOV R7,#20
exitTimer0:
     RETI
Inc_clock:
     INC Second              ; Tăng giây
     MOV A,Second
     CJNE A,#60,exitInc      ; Nếu giây < 60 thì thoát
     MOV Second,#0           ; Ngược lại thì gán giây = 0
     INC Minute              ; và tăng phút
     MOV A,Minute
     CJNE A,#60,exitInc      ; Nếu phút < 60 thi thoát
     MOV Minute,#0           ; Ngược lại thì gán phút = 0
     INC Hour                ; và tăng giờ
     MOV A,Hour
     CJNE A,#24,exitInc      ; Nếu giờ < 24 thì thoát
     MOV Hour,#0             ; Ngược lại thì gán giờ = 0
exitInc:
     RET
     END
```

3.4. Ngắt do cổng nối tiếp

MCS-51 có 2 nguồn ngắt do cổng nối tiếp: ngắt phát và ngắt thu. Hai nguồn ngắt này xác định bằng các bit RI, TI và dùng chung một địa chỉ ISR nên khi chuyển đến ISR, các cờ ngắt không tự động xoá bằng phần cứng mà phải thực hiện bằng phần mềm: kiểm tra nguyên nhân ngắt (RI hay TI) và xoá bit cờ tương ứng.

Ví dụ: Viết chương trình khởi động cổng nối tiếp ở chế độ UART 8 bit với tốc độ truyền 4800 bps. Viết ISR cho cổng nối tiếp theo yêu cầu: truyền tuần tự các ký tự từ 'A' đến 'Z' ra cổng nối tiếp đồng thời mỗi lần có ký tự đến cổng nối tiếp thì nhận về và xuất ký tự nhận ra P0 (giả sử tần số thạch anh là 11.0592 MHz).

Giải

- Nội dung thanh ghi SCON:

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
0	1	0	1	0	0	0	0
UART 8 bit		Không ở chế độ đa xử lý	Cho phép thu			Không cho phép truyền	

SCON = 50h

- Nội dung thanh ghi TMOD:

GATE1	C/T1	M11	M10	GATE0	C/T0	M01	M00
0	0	1	0	0	0	0	0
Không dùng INT1	Đếm bằng dao động nội	Chế độ 8 bit		Timer 0 không dùng			

TMOD = 0010 0000b (20h)

- Giá trị đếm (theo bảng 3.10): TH1 = -6

- Nội dung thanh ghi IE:

EA	-	ET2	ES	ET1	EX1	ET0	EX0
1	0	0	1	0	0	0	0

IE = 1001 0000b (90h)

Chương trình thực hiện như sau:

```
     ORG 0000h
     LJMP main
     ORG 0023h     ; Địa chỉ ISR của cổng nối tiếp
     LJMP Serial_ISR
Main:
     MOV TMOD,#20h
     MOV TH1,#(-6)
     MOV TL1,#(-6) ; Tốc độ 4800 bps
     SETB TR1
     MOV R7,#'A'   ; Ký tự truyền đầu tiên
     MOV IE,#90h   ; Cho phép ngắt tại công nối tiếp
     SETB TI       ;Cho phép truyền
     SJMP $
```

```
Serial_ISR:
     JNB RI,Transmit    ; Nếu không phải ngắt do nhận
                        ; ký tự thì truyền
     CLR RI
     MOV A,SBUF         ; Nhận ký tự
     MOV P0,A           ; Xuất ra Port 0
     SJMP exitSerial
Transmit:               ; Truyền ký tự
     CLR TI
     MOV A,R7
     MOV SBUF,A         ; Truyền ký tự
     INC R7             ; Qua ký tự kế
     CJNE R7,#'Z'+1,exitSerial   ; Nếu chưa truyền'Z' thì
                        ; tiếp tục truyền, ngược lại thì
     MOV R7,#'A'        ; bắt đầu truyền từ ký tự 'A'
exitSerial:
     RETI
     END
```

3.5. Ngắt ngoài

MCS-51 có 2 nguồn ngắt ngoài khác nhau: ngắt ngoài 0 và ngắt ngoài 1. Ngắt ngoài xảy ra khi bit IEx chuyển lên mức 1, quá trình chuyển mức của bit IEx xảy ra khi:

- Bit ITx = 0 và xuất hiện mức logic 0 tại chân INTx tương ứng (P3.2 cho ngắt ngoài 0 hay P3.3 cho ngắt ngoài 1).
- Bit ITx = 1 và xuất hiện cạnh âm tại chân INTx.

Khi có ngắt xảy ra và cho phép ngắt (dùng thanh ghi IE), chương trình sẽ được chuyển đến địa chỉ của ISR tương ứng (0003h cho ngắt ngoài 0 và 0013h cho ngắt ngoài 1) và xoá cờ ngắt TFx.

Lưu ý rằng các cờ ngắt được lấy mẫu trong mỗi chu kỳ nên để phát hiện ngắt, yêu cầu phải:

- Ở mức thấp tối thiểu 1 chu kỳ nếu tác động bằng mức logic (ITx = 0).
- Ở mức cao tối thiểu 1 chu kỳ trước khi chuyển xuống mức thấp và mức thấp cũng phải tồn tại tối thiểu 1 chu kỳ (ITx = 1).

Quá trình điều khiển ngắt ngoài mô tả như sau:

- Xác định yêu cầu ngắt bằng cạnh âm hay bằng mức logic.

- Cho phép ngắt tại ngắt ngoài tương ứng (dùng thanh ghi IE).
- Xác định mức ưu tiên (thanh ghi IP).
- Viết ISR cho các ngắt.

Ví dụ: Viết chương trình sao cho mỗi khi có mức logic 0 xuất hiện tại P3.2 (ngắt ngoài 0) thì tạo xung 1 KHz tại P1.0. Quá trình tạo xung chỉ dừng khi có mức logic 0 xuất hiện tại P3.3 (ngắt ngoài 1).

Giải

Chương trình thực hiện có 3 ngắt xảy ra: ngắt ngoài 0 cho phép timer chạy để tạo xung tại P1.0, ngắt ngoài 1 cấm timer để ngừng quá trình tạo xung và ngắt timrer để tạo xung.

f = 1 KHz → T = 1ms (1000 chu kỳ): giá trị đếm là 500 (chế độ 16 bit)

- Nội dung thanh ghi TMOD:

GATE1	C/T1	M11	M10	GATE0	C/T0	M01	M00
0	0	0	1	0	0	0	0
Không dùng INT1	Đếm bằng dao động nội	Chế độ 16 bit		Timer 0 không dùng			

TMOD = 0001 0000b (10h)

- Nội dung thanh ghi IE:

EA	-	ET2	ES	ET1	EX1	ET0	EX0
1	0	0	0	1	1	0	1

IE = 1000 1101b (8Dh)

Chương trình thực hiện như sau:

```
ORG 0000h
LJMP main

ORG 0003h          ; Địa chỉ ISR ngắt ngoài 0
SETB TR1           ; Timer 1 chạy
RETI

ORG 0013h          ; Địa chỉ ISR của ngắt ngoài 1
CLR TR1            ; Cấm timer 1
RETI

ORG 001Bh          ; Địa chỉ ISR timer 1
MOV TH1,#HIGH(-500); Chế độ 16 bit nên mỗi lần tràn
MOV TL1,#LOW(-500); phải nạp lại giá trị
CPL P1.0           ; Đảo bit P1.0 để tạo xung
RETI
```

```
Main:
     MOV TMOD,#10h
     MOV TH1,#HIGH(-500)
     MOV TL1,#LOW(-500)
     MOV IE,#8Dh   ; Cho phép ngắt tại ngắt ngoài 0, 1 và
     SJMP $        ; timer 1
     END
```

BÀI TẬP CHƯƠNG 3

1. Viết đoạn chương trình theo yêu cầu:

- Khởi động cổng nối tiếp ở chế độ UART 8 bit với tốc độ truyền 4800 bps.
- Định thời 1s thì đọc dữ liệu từ P1, lưu vào ô nhớ 30h và xuất dữ liệu vừa đọc ra cổng nối tiếp.

2. Viết đoạn chương trình theo yêu cầu:

- Khởi động cổng nối tiếp ở chế độ UART 9 bit với tốc độ truyền 9600 bps.
- Khi có ngắt xảy ra tại ngắt ngoài 0 thì xuất dữ liệu tại ô nhớ 30h ra cổng nối tiếp trong đó bit truyền thứ 9 là bit parity.
- Khi có ngắt tại ngắt ngoài 1 thì đọc dữ liệu từ P0 và lưu kết quả vào ô nhớ 30h.

3. Viết đoạn chương trình theo yêu cầu:

- Khi có ngắt tại ngắt ngoài 0 (tác động bằng cạnh) thì đọc dữ liệu tại P2 và lưu vào ô nhớ 30h đồng thời tăng giá trị trong ô nhớ lên 1.
- Dùng ngắt timer 0 định thời 30s thì đọc giá trị trong ô nhớ 30h, xoá nội dung trong ô nhớ 31h và kiểm tra giá trị theo yêu cầu:

Giá trị	**Thực hiện**
> 200	Đặt bit P1.0 = 1, xoá bit P1.1 = 0 và P1.2 = 0 Tạo xung f = 1KHz tại P1.3 dùng ngắt timer 1
< 100	Đặt bit P1.1 = 1, xoá bit P1.0 = 0 và P1.2 = 0 Ngừng tạo xung tại P1.3
Khác	Đặt bit P1.2 = 1, xoá bit P1.0 = 0 và P1.1 = 0