ریزپردازنده

دانشکده کامپیوتر دانشگاه یزد

نیم‌سال دوم تحصیلی ۹۷-۹۶

ارائه‌دهنده : پریسا استواری

روش کار تایمر

تایمر

- تایمر یا شمارنده تشکیل شده است از یک یا چند رجیستر.
- تایمر برای اندازه‌گیری زمان یا تولید تاخیر زمانی استفاده می‌شود.
  - برای مثال اگر تایمر ۸ بیتی باشد و مقدار 201 در رجیستر آن باشد، تایمر از 201 با فرکانس تایمر شروع به شمارش می‌کند و در هر کلاک پالس یک واحد به مقدار رجیستر آن اضافه می‌گردد.
  - هنگامی که مقدار رجیستر برابر با 255 شد و در پالس بعدی مقدار آن 0 گردید، سرریز رخ داده و مقدار یک پرچم برابر با یک می‌شود.
  - با چک کردن این پرچم می‌توان فاصله‌های زمانی را اندازه گرفت. برای محاسبه زمان سپری شده دانستن دو چیز مورد نیاز است.
    - فرکانس تایمر
    - مقدار اولیه تایمر
  - در مثال بالا اگر فرکانس تایمر 1MHz باشد یعنی در هر 1µs یک واحد به تایمر اضافه می‌شود. برای اینکه تایمر از مقدار 201 به 255 رسیده و در شمارش بعد 0 شود، باید 55 بار افزایش یابد. پس از زمانی که تایمر شروع به شمارش کرد تا زمانی که پرچم آن یک شود 55µs طول خواهد کشید.

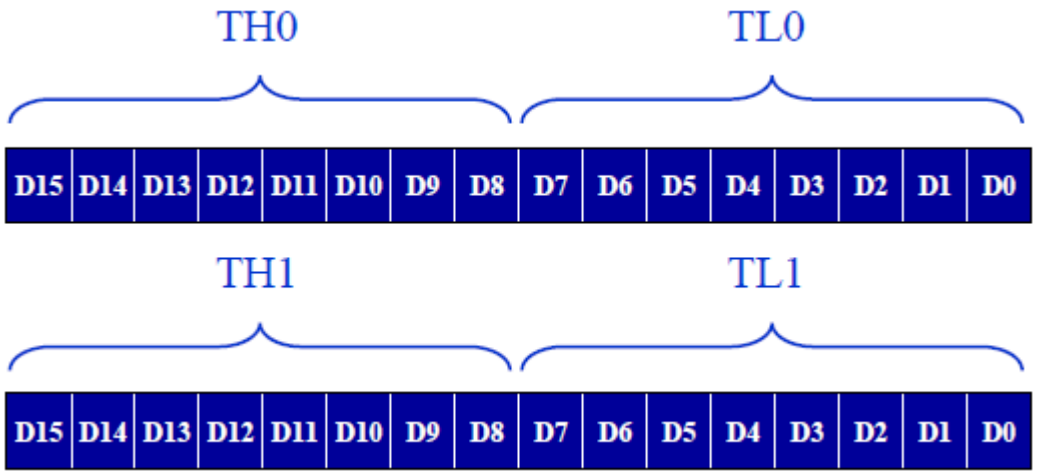
شمارنده

- تایمر یا شمارنده تشکیل شده است از یک یا چند رجیستر.
  - تایمر و شمارنده در واقع دارای یک سخت‌افزار هستند.

- شمارنده برای شمارش اتفاقاتی که در بیرون از میکروکنترلر می‌افتد استفاده می‌شود.
  - یک اتفاق یعنی یک تغییر از ۱ به ۰ در یکی از پایه‌های میکروکنترلر (مثلا پایه T0 یا T1)
  - فرض کنید پایه T0 به یک دستگاه وصل باشد. تا زمانی که دستگاه روشن باشد مقدار پایه T0 یک خواهد بود. به محض خاموش شدن دستگاه مقدار این پایه صفر خواهد شد.
  - در واقع در لحظه‌ی خاموش شدن دستگاه یک لبه‌ی پایین رونده روی پایه T0 ایجاد می‌شود که باعث مقدار شمارنده یک واحد افزایش یابد.
  - اگر مقدار شمارنده در ابتدا صفر باشد، هر بار که دستگاه خاموش شود یک لبه‌ی پایین رونده در پایه T0 ایجاد شده و یک واحد به شمارنده اضافه می‌شود.
  - پس در هر زمان، مقدار شمارنده تعداد دفعات خاموش شدن دستگاه را نشان می‌دهد. (شمارش اتفاقات خارجی)

رجیستر تایمر0 و 1

- میکروکنترلر 8051 دارای ۲ تایمر/شمارنده است.
  - در این اسلاید به جای نام تایمر/شمارنده از نام تایمر استفاده می‌کنیم.
- 8051 دارای دو تایمر 16 بیتی است به نام‌های تایمر 0 و تایمر 1
- از آنجایی که 8051 دارای معماری 8 بیتی است، برای دسترسی به تایمر 16 بیتی از دو رجیستر 8 بیتی استفاده می‌شود.
  - رجیستر کم ارزش‌تر به نام TL0 و TL1 برای تایمر 0 و تایمر 1 است.
  - رجیستر پر ارزش‌تر به نام TH0 و TH1 است.
  - می‌توان به آنها مانند تمام رجیسترها دسترسی داشت.
    - MOV TH0, #4AH
    - MOV R5, TL1



ثبات‌های مورد استفاده تایمرها

- در 8051 تایمرها از ۶ ثبات کاربرد خاص (SFR) استفاده می‌کنند.

Timer special function registers

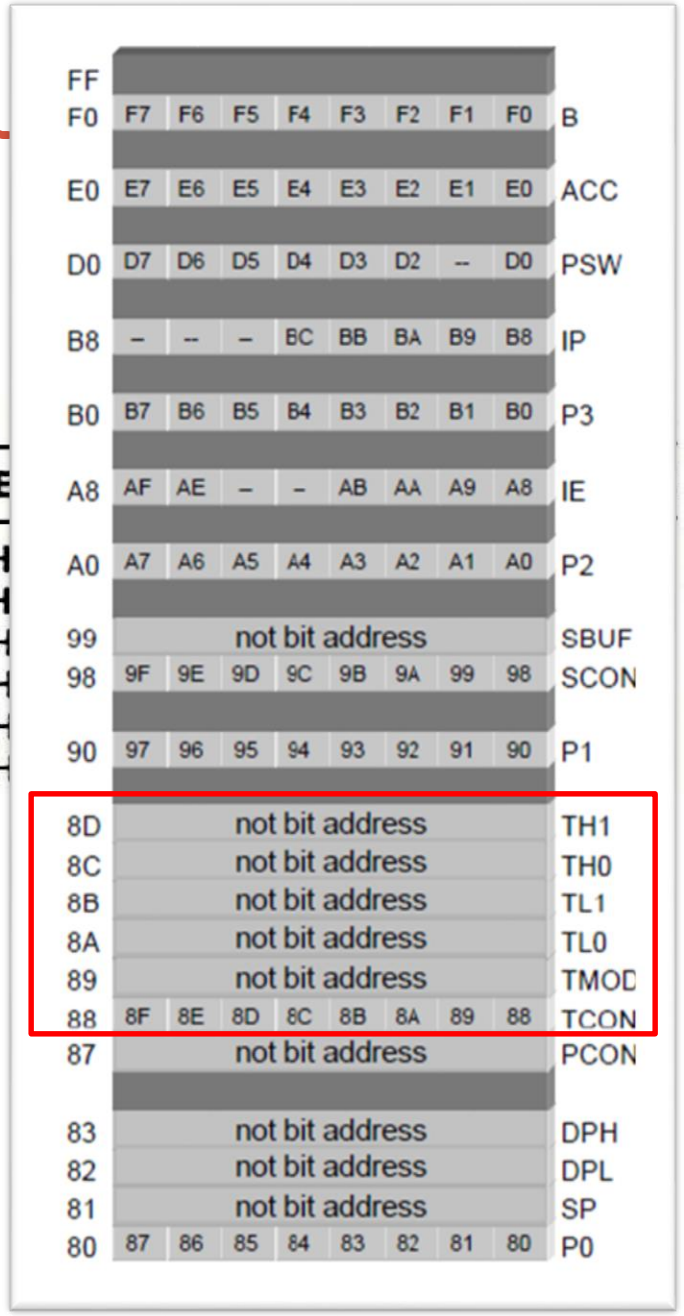
TIMER SFR	PURPOSE	ADDRESS	BIT-ADDRESSABLE
TCON	Control	88H	Yes
TMOD	Mode	89H	No
TL0	Timer 0 low-byte	8AH	No
TL1	Timer 1 low-byte	8BH	No
TH0	Timer 0 high-byte	8CH	No
TH1	Timer 1 high-byte	8DH	No

ساده تایمرها

کاربرد خاص (SFR) استفاده می‌کنند.

Timer special function registers

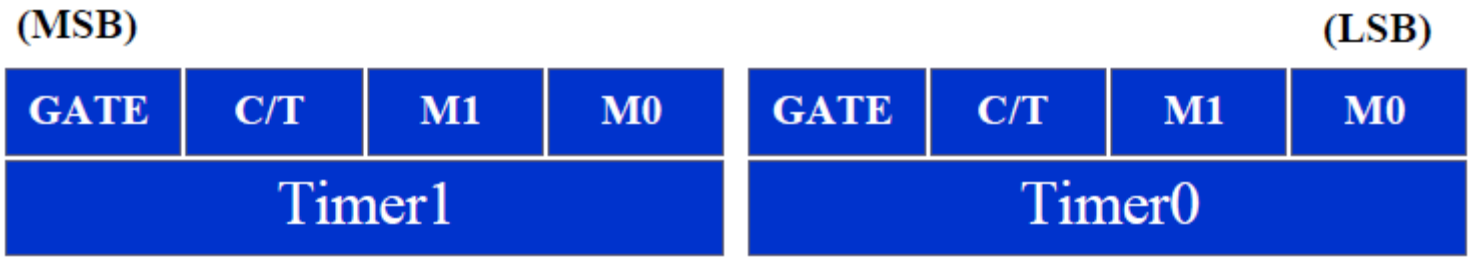
TIMER SFR	PURPOSE	ADDRE
TCON	Control	88H
TMOD	Mode	89H
TL0	Timer 0 low-byte	8AH
TL1	Timer 1 low-byte	8BH
TH0	Timer 0 high-byte	8CH
TH1	Timer 1 high-byte	8DH

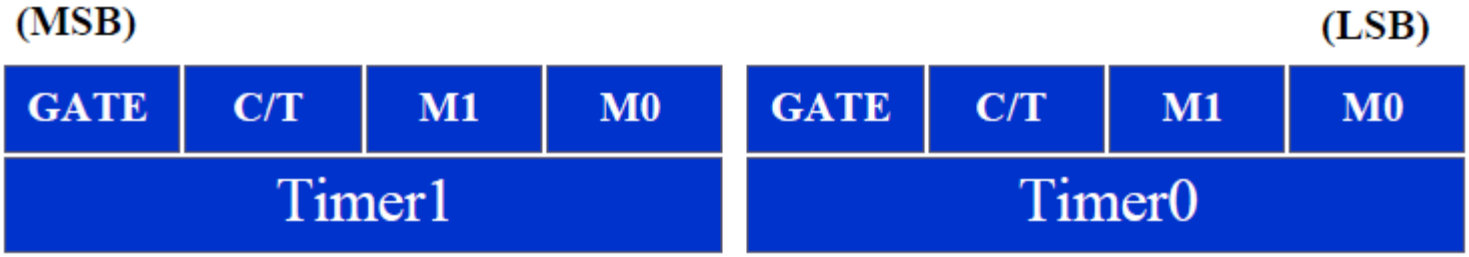


رجیستر TMOD

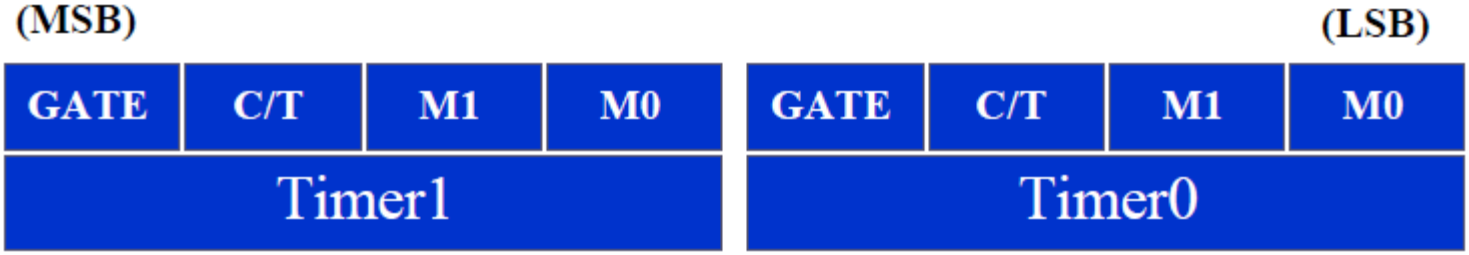
- تعیین مود عملکرد هر دو تایمر 0 و 1 از طریق رجیستر TMOD صورت می‌گیرد.
  - TMOD (Timer Mode)

- TMOD یک رجیستر 8 بیتی است.
  - ۴ بیت کم ارزش‌تر آن برای تایمر 0
  - ۴ بیت پر ارزش‌تر آن برای تایمر 1
  - برای هر کدام از تایمرها :
    - ۲ بیت کوچکتر برای تعیین مود تایمر به کار می‌رود.
    - ۲ بیت بزرگتر برای مشخص کردن عملیات.





شماره بیت	نام	شماره تایمر	عملکرد
7	GATE	1	بیت گیت زمانی که ۱ می‌شود، تایمر به شرطی کار می‌کند که $\overline{INT1}$ برابر ۱ باشد.
6	C/$\overline{T}$	1	اگر ۰ باشد تایمر 1 به عنوان تایمر کار می‌کند. اگر ۱ باشد به عنوان شمارنده کار می‌کند.
5	M1	1	بیت ۱ حالت تایمر 1
4	M0	1	بیت ۰ حالت تایمر 1
3	GATE	0	بیت گیت زمانی که ۱ شود، تایمر 0 به شرطی کار می‌کند که $\overline{INT0}$ برابر با ۱ باشد.
2	C/$\overline{T}$	0	اگر ۰ باشد تایمر 0 به عنوان تایمر کار می‌کند. اگر ۱ باشد به عنوان شمارنده کار می‌کند.
1	M1	0	بیت ۱ حالت تایمر 0
0	M0	0	بیت ۰ حالت تایمر 0



شماره بیت	نام	شماره تایمر	عملکرد
7	GATE	1	بیت گیت زمانی که ۱ می‌شود، تایمر به شرطی کار می‌کند که $\overline{INT1}$ برابر ۱ باشد.
6	C/$\overline{T}$	1	اگر ۰ با اگر ۱ با
5	M1	1	بیت ۱
4	M0	1	بیت ۰
3	GATE	0	بیت گیت زمانی که ۱ شود، تایمر 0 به شرطی کار می‌کند که $\overline{INT0}$ برابر با ۱ باشد.
2	C/$\overline{T}$	0	اگر ۰ باشد تایمر 0 به عنوان تایمر کار می‌کند. اگر ۱ باشد به عنوان شمارنده کار می‌کند.
1	M1	0	بیت ۱ حالت تایمر 0
0	M0	0	بیت ۰ حالت تایمر 0

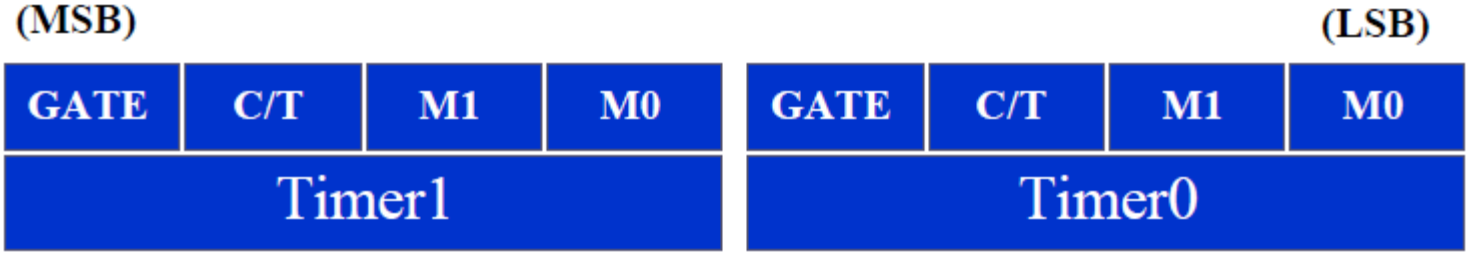
تایمر یا شمارنده تنها در حالتی فعال خواهند شد که هم پایه‌ی $\overline{INTx}$ یک باشد، هم بیت TRx یک باشد.
یعنی شروع به کار و متوقف شدن تایمر از بیرون از میکروکنترلر کنترل خواهد شد.

(MSB) (LSB)

GATE	C/T	M1	M0	GATE	C/T	M1	M0
				ner0			

انتخاب تایمر یا شمارنده
صفر انتخاب حالت تایمر (کلاک تایمر را کلاک میکروکنترلر تامین می‌کند)
یک انتخاب حالت شمارنده (کلاک شمارنده از پین ورودی T0 یا T1 تامین می‌شود)

			بیت گیت زمانی که ۱ می‌شود، تایمر به
6	C/$\overline{T}$	1	اگر ۰ باشد تایمر 1 به عنوان تایمر کار می‌کند. اگر ۱ باشد به عنوان شمارنده کار می‌کند.
5	M1	1	بیت ۱ حالت تایمر 1
4	M0	1	بیت ۰ حالت تایمر 1
3	GATE	0	بیت گیت زمانی که ۱ شود، تایمر 0 به شرطی کار می‌کند که $\overline{INT0}$ برابر با ۱ باشد.
2	C/$\overline{T}$	0	اگر ۰ باشد تایمر 0 به عنوان تایمر کار می‌کند. اگر ۱ باشد به عنوان شمارنده کار می‌کند.
1	M1	0	بیت ۱ حالت تایمر 0
0	M0	0	بیت ۰ حالت تایمر 0



شماره بیت	نام	شماره تایمر	عملکرد
7	GATE	1	بیت گیت زمانی که ۱ می‌شود، تایمر به شرطی کار می‌کند که $\overline{INT1}$ برابر ۱ باشد.
6	C/$\overline{T}$	1	اگر ۰ باشد تایمر 1 به عنوان تایمر کار می‌کند. اگر ۱ باشد به عنوان شمارنده کار می‌کند.
5	M1	1	بیت ۱ حالت تایمر 1
4	M0	1	بیت ۰ حالت تایمر 1
3	GATE		بیت گیت زمانی که ۱ شود، تایمر 0 به
2	C/$\overline{T}$		اگر ۰ باشد تایمر 0 به عنوان تایمر کار اگر ۱ باشد به عنوان شمارنده کار می‌کند.
1	M1	0	بیت ۱ حالت تایمر 0
0	M0	0	بیت ۰ حالت تایمر 0

با استفاده از این دو بیت می‌توان هر یک از تایمرها را در ۴ وضعیت مختلف قرار داد.

رجیستر TMOD

- دقت شود، بیت‌های ثبات TMOD را نمی‌توان به صورت مجزا مقداردهی کرد.

- معمولا در ابتدای برنامه یک بار ثبات TMOD مقداردهی می‌شود و طرز کار تایمرها مشخص می‌شود.
- سپس در طول برنامه، با مقداردهی به ثبات‌های کاربرد خاص دیگر مخصوصا ثبات TCON تایمر راه‌اندازی یا متوقف می‌شود.

رجیستر TCON

- برای بررسی وضعیت یا کنترل تایمرها استفاده می‌شود.

- یک ثبات ۸ بیتی است که بیت‌های آن آدرس پذیراند.
  - چهار بیت بالای آن (۴ تا ۷) برای راه‌اندازی یا توقف تایمرها (TR0 و TR1) و چک کردن سرریز در تایمرها (TF0 و TF1) است.
  - چهار بیت پایین آن برای تشخیص و ایجاد وقفه‌ی خارجی (اسلاید ۱۲) است.

TCON (timer control) register summary

BIT	SYMBOL	BIT ADDRESS	DESCRIPTION
TCON.7	TF1	8FH	Timer 1 overflow flag. Set by hardware upon overflow; cleared by software, or by hardware when processor vectors to interrupt service routine
TCON.6	TR1	8EH	Timer 1 run-control bit. Set/cleared by software to turn timer on/off
TCON.5	TF0	8DH	Timer 0 overflow flag
TCON.4	TR0	8CH	Timer 0 run-control bit
TCON.3	IE1	8BH	External interrupt 1 edge flag. Set by hardware when a falling edge is detected on $\overline{\text{INT 1}}$; cleared by software, or by hardware when CPU vectors to interrupt service routine
TCON.2	IT1	8AH	External interrupt 1 type flag. Set/cleared by software for falling edge/low-level activated external interrupt
TCON.1	IE0	89H	External interrupt 0 edge flag
TCON.0	IT0	88H	External interrupt 0 type flag

TCON (timer control) register summary

BIT	SYMBOL	BIT ADDRESS	DESCRIPTION
TCON.7	TF1	8FH	Timer 1 overflow flag. Set by hardware upon overflow; cleared by software, or by hardware when processor vectors to interrupt service routine
TCON.6	TR1	8EH	Timer 1 run-control bit. Set/cleared by software to turn timer on/off
TCON.5	TF0	8DH	Timer 0 overflow flag
TCON.4	TR0	8CH	Timer 0 run-control bit
TCON.3	IE1	8BH	External interrupt 1 edge flag. Set by hardware when edge is detected on
TCON.2	IT1	8AH	Interrupt 1 type flag. Set/cleared by software for edge/low-level activated
TCON.1	IE0	89H	External interrupt 0 edge flag
TCON.0	IT0	88H	External interrupt 0 type flag

موقعی که تایمر پر شد بیت سرریز (TFx) تایمر برابر ۱ می‌شود. (Timer Flag 0/1)
می‌توان در برنامه این بیت را چک کرد یا مقدار آن را صفر نمود.
به صورت سخت‌افزاری نیز زمانی که پروسسور وارد روتین وقفه می‌شود، این بیت ۰ می‌شود. (اسلاید ۱۲)

TCON (timer control) register summary

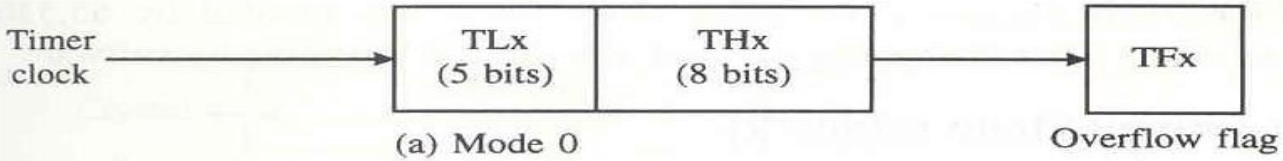
BIT	SYMBOL	BIT ADDRESS	DESCRIPTION
TCON.7	TF1	8FH	Timer 1 overflow flag. Set by hardware upon overflow; cleared by software, or by hardware when processor vectors to interrupt service routine
TCON.6	TR1	8EH	Timer 1 run-control bit. Set/cleared by software to turn timer on/off
TCON.5	TF0	8DH	Timer 0 overflow flag
TCON.4	TR0	8CH	Timer 0 run-control bit
TCON.3	IE1	8BH	External interrupt 1 edge flag. Set by hard- detected on
TCON.2			r by hardware upt service Set/cleared by software for falling edge/low-level activated external interrupt
TCON.1	IE0	89H	External interrupt 0 edge flag
TCON.0	IT0	88H	External interrupt 0 type flag

این بیت برای راه‌اندازی یا توقف تایمرها استفاده می‌شود و به صورت نرم‌افزاری و از طریق برنامه ۰ یا ۱ می‌شود.
زمانی که این بیت ۱ شود، تایمر مربوطه شروع به کار می‌کند.
زمانی که این بیت ۰ شود، تایمر مربوطه متوقف می‌شود.

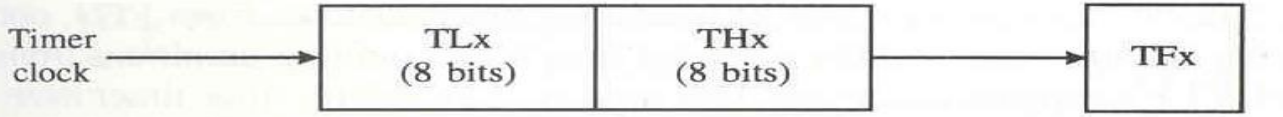
حالت یا مدهای مختلف تایمر

- با استفاده از بیت‌های M0 و M1 در ثبات TMOD می‌توان ۴ حالت برای تایمر/شمارنده‌ها ایجاد نمود.

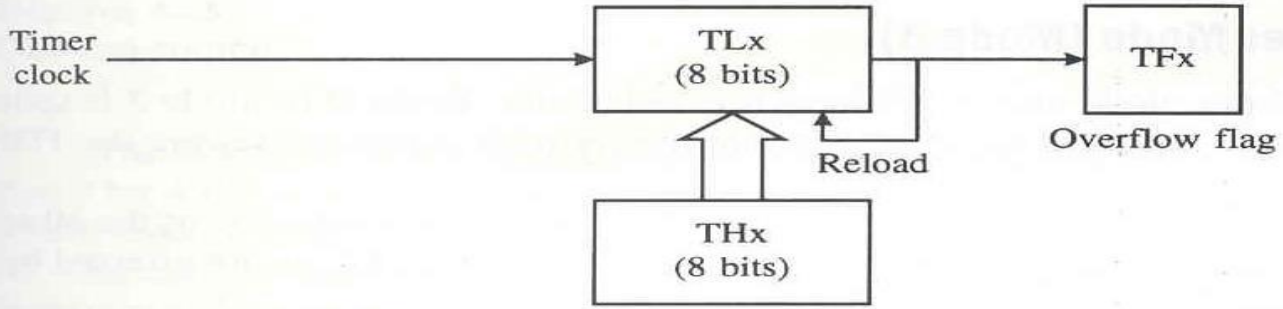
M1	M0	حالت تایمر	عملکرد تایمر
0	0	0	حالت تایمر/شمارنده ۱۳ بیتی
0	1	1	حالت تایمر/شمارنده ۱۶ بیتی
1	0	2	حالت تایمر/شمارنده ۸ بیتی با بارشدن خودکار
1	1	3	در این حالت تایمر 0 دو قسمت می‌شود یعنی دو تایمر مجزای ۸ بیتی در TL0 و TH0 داریم. تایمر 1 متوقف می‌شود.



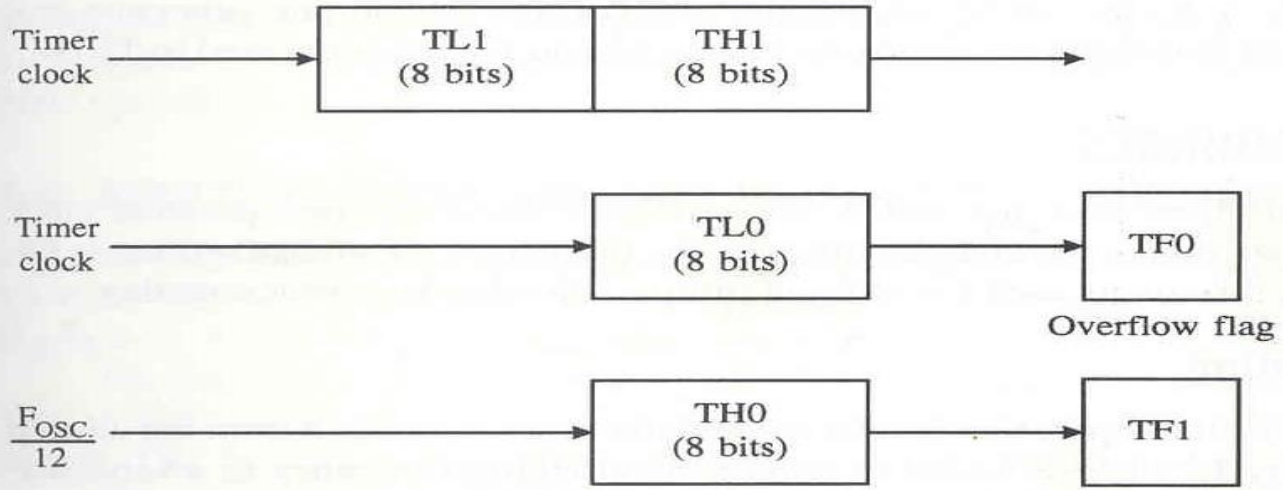
(a) Mode 0



(b) Mode 1



(c) Mode 2



(d) Mode 3

FIGURE 4–2
Timer modes (a) Mode 0 (b) Mode 1 (c) Mode 2 (d) Mode 3

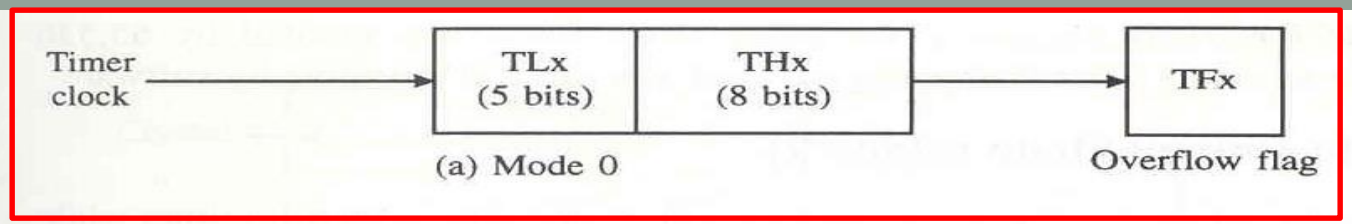
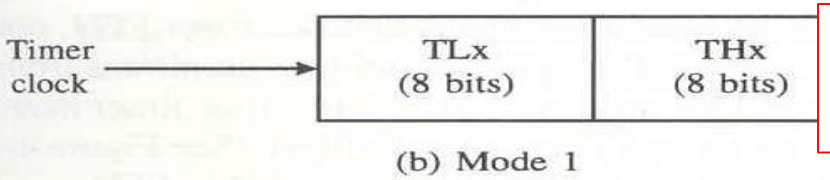
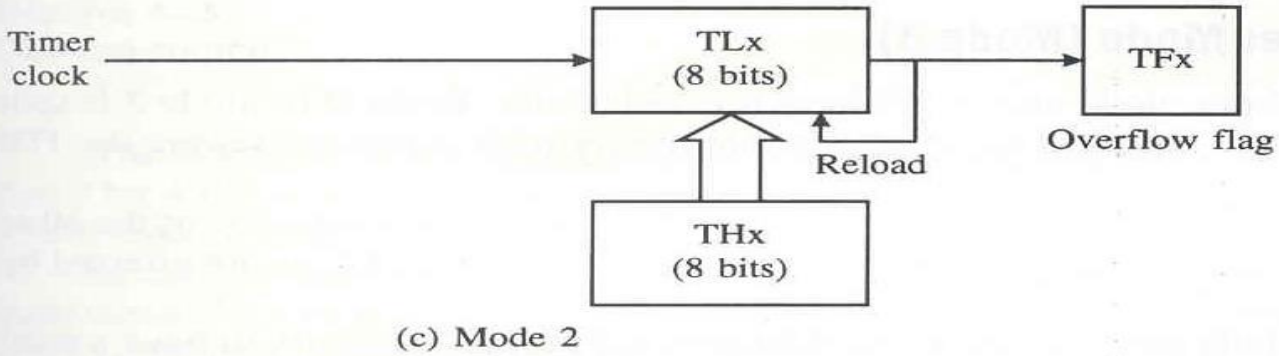
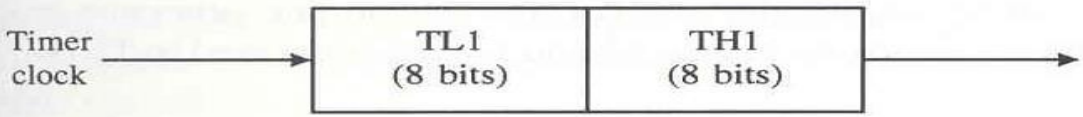


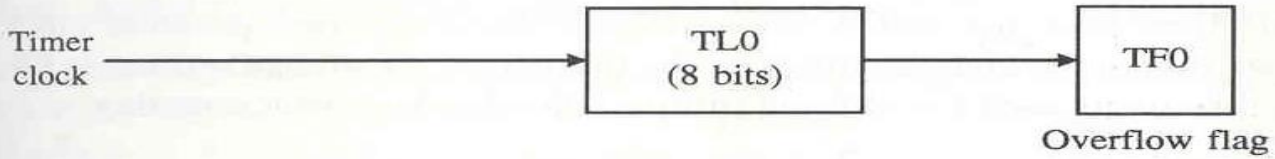
FIGURE 4–2
Timer modes (a) Mode 0 (b) Mode 1 (c) Mode 2 (d) Mode 3

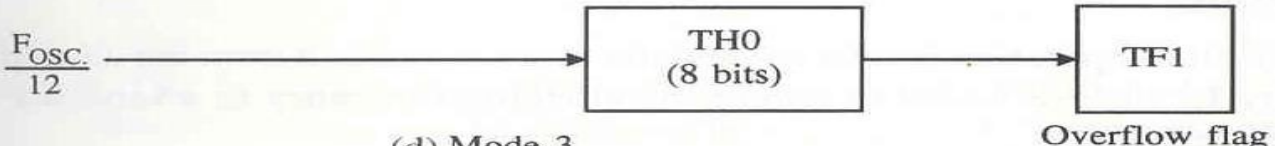


مود 0 = تایمر ۱۳ بیتی
این حالت در طرح‌های جدید به کار برده نمی‌شود.









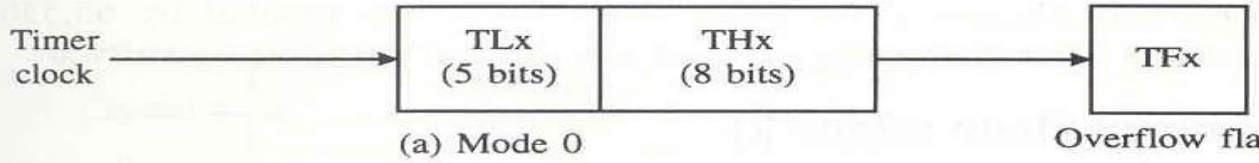
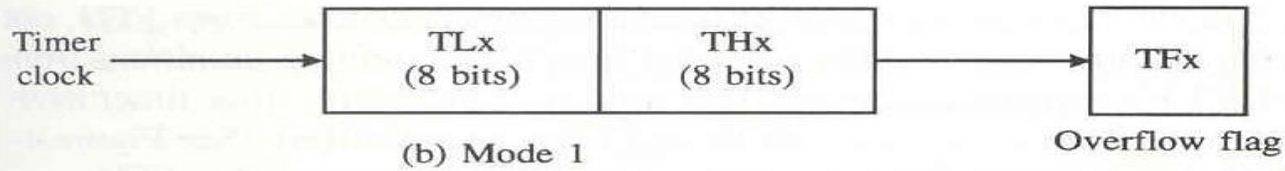
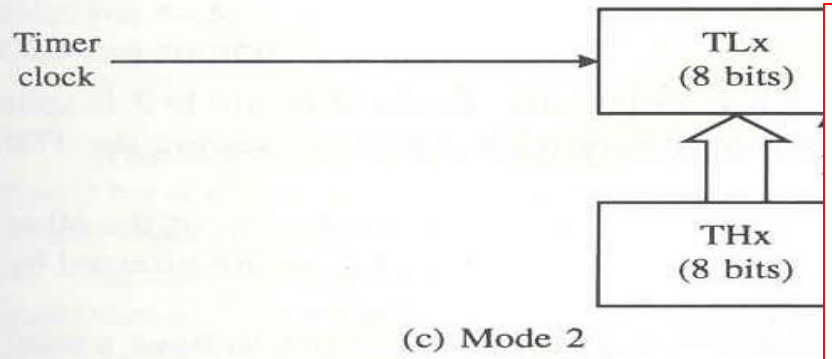
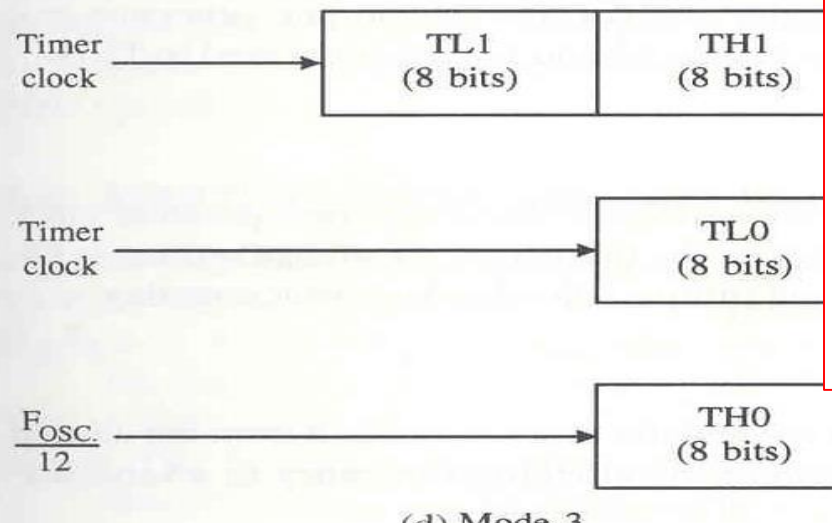


FIGURE 4–2
Timer modes (a) Mode 0 (b) Mode 1 (c) Mode 2 (d) Mode 3







مود 1 = تایمر ۱۶ بیتی

دو ثبات TLx و THx در این شمارش نقش دارند.
زمانی که مقدار این دو ثبات به FFFFH (۶۵۵۳۵) رسید و در پالس بعدی 0000H شد، پرچم TFx یک می‌شود و تایمر همچنان به شمارش خود از صفر ادامه می‌دهد.

پرچم سرریز TFx را می‌توان با برنامه خواند یا مقدار آن را عوض نمود.

در هر لحظه می‌توان با برنامه محتوای ثبات‌های TLx و THx را خواند یا بر روی آنها نوشت.

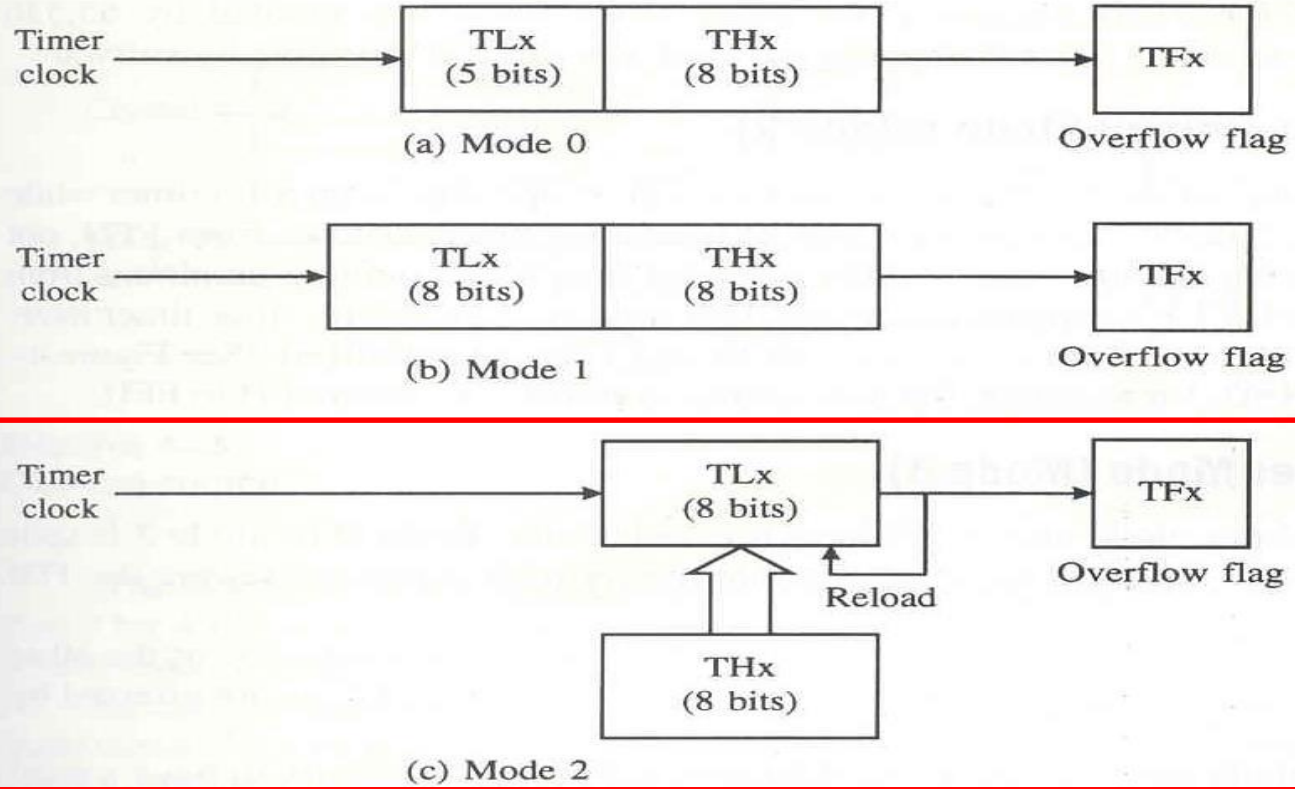


FIGURE 4–2
Timer modes (a) Mode 0 (b) Mode 1 (c) Mode 2 (d) Mode 3

مود 2 = تایمر ۸ بیتی با بار شدن خودکار

بایت بزرگتر تایمر (THx) عددی که باید هر بار در تایمر بار شود را نگه می‌دارد.
هنگامی که TLx تا FFH (۲۵۵) شمرد و سپس 00H شد، بیت پرچم سرریز TFx یک می‌شود و مقداری که در بایت بزرگتر (THx) نگهداری شده، به صورت خودکار به تایمر (TLx) بار می‌شود و عمل شمارش ادامه می‌یابد.
دقت شود کافی است THx فقط یک بار مقداردهی شود.

در این حالت، سرریز در فواصل زمانی مشخص و برابر اتفاق می‌افتد.

مثال : اگر مقدار THx برابر با 4FH باشد، شمارش از 4FH تا FFH ادامه می‌یابد، سپس سرریز رخ می‌دهد و TFx یک می‌شود و باز شمارش از 4FH ادامه می‌یابد.

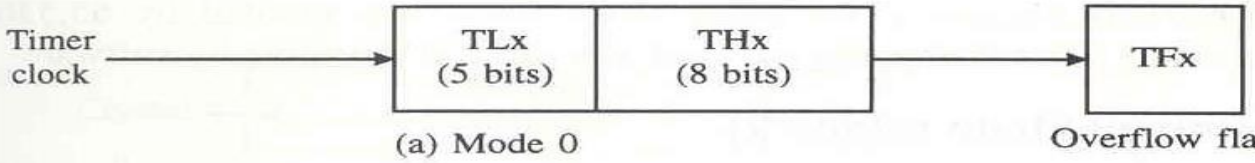
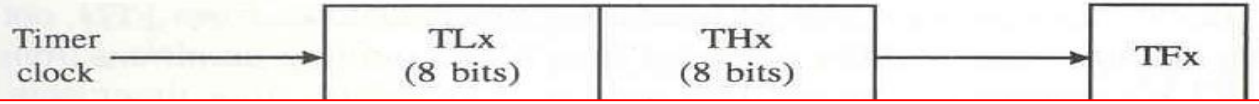


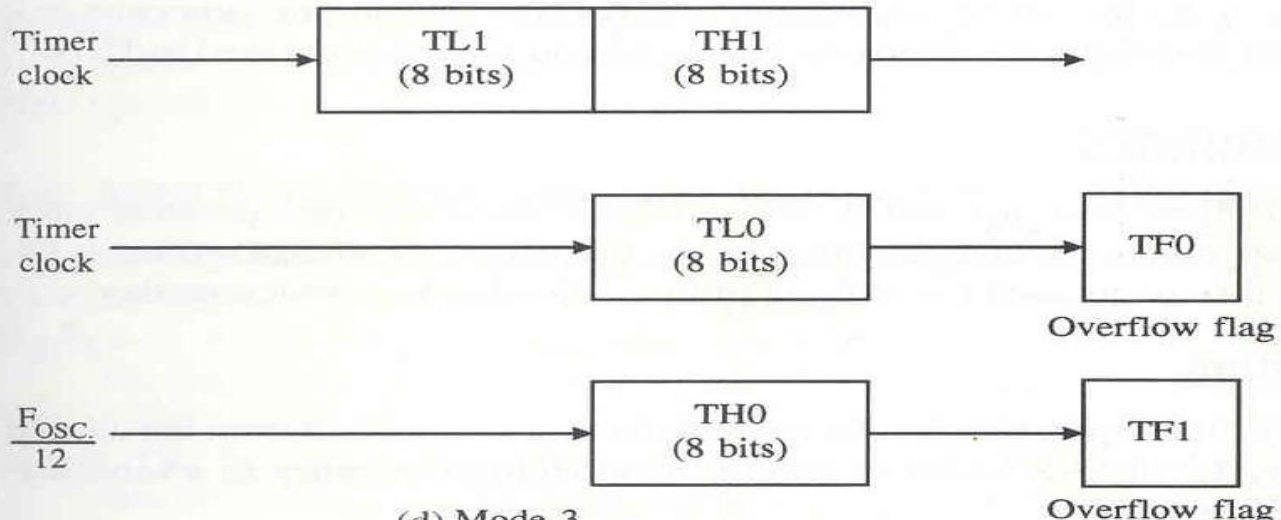
FIGURE 4–2
Timer modes (a) Mode 0 (b) Mode 1 (c) Mode 2 (d) Mode 3



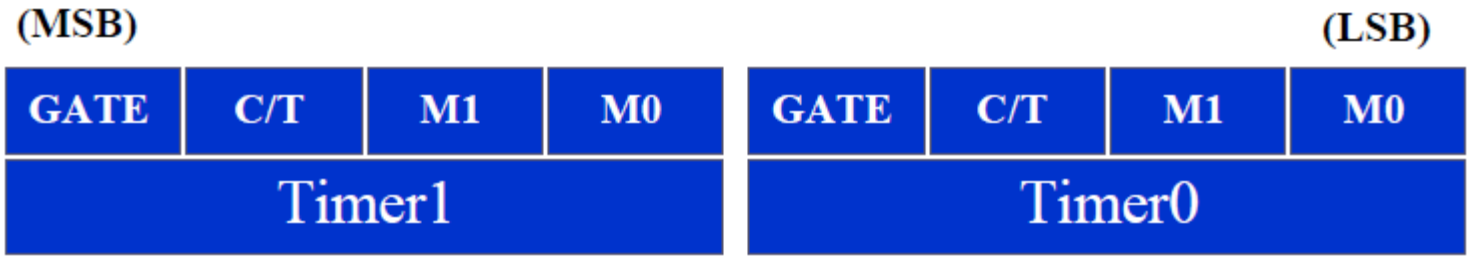
مود 3 = حالت تفکیک شده (Split Timer Mode)

تایمر 0 به دو تایمر مجزا تفکیک می‌شود.
TL0 یک تایمر ۸ بیتی با پرچم سرریز TF0 و TH0 یک تایمر ۸ بیتی با پرچم سرریز TF1

تایمر 1 متوقف می‌شود. اما می‌توان آن را به مودهای دیگر تغییر داد و شروع به کار نماید. از آنجایی که تایمر 1 پرچم سرریز ندارد، می‌توان آن را برای کارهایی که نیاز به وقفه ندارد مثلا تولید پالس ساعت در پورت سری (Baud Rate Clock) تنظیم کرد.



مثال



- مشخص کنید در هر حالت چه تایمری و چه مودی فعال شده است؟

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

Solution:

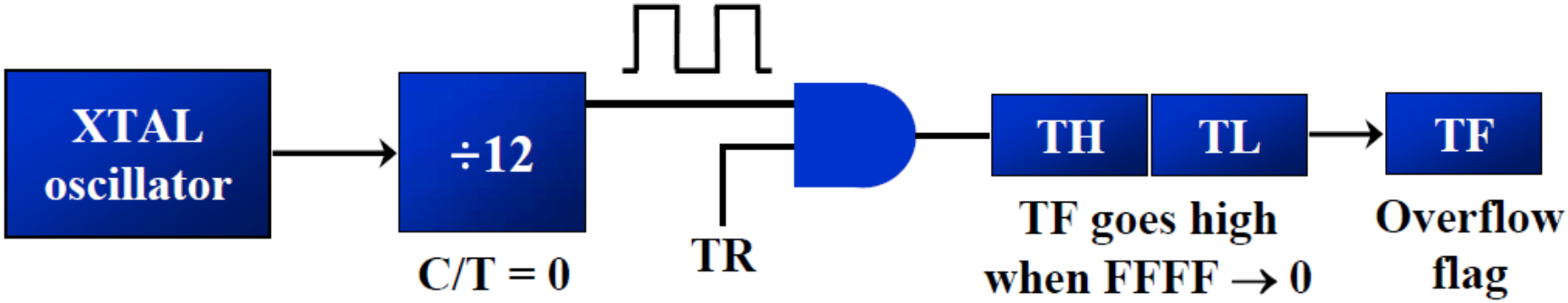
We convert the value from hex to binary. From Figure 9-3 we have:
(a) TMOD = 00000001, mode 1 of timer 0 is selected.
(b) TMOD = 00100000, mode 2 of timer 1 is selected.
(c) TMOD = 00010010, mode 2 of timer 0, and mode 1 of timer 1 are selected.

منابع پالس ساعت

- برای تایمرها دو نوع منبع پالس وجود دارد که با $C/\overline{T}$ انتخاب می‌شود.

۱. یک منبع پالس برای اندازه‌گیری زمان

- اگر بیت $C/\overline{T}$ در ثبات TMOD صفر باشد، تایمر پالس خود را از اسیلاتور تراشه تامین می‌کند.
- یک مدار برای کاهش فرکانس پالس ساعت وجود دارد که فرکانس پالس اسیلاتور را بر ۱۲ تقسیم می‌کند.
- برای مثال اگر فرکانس اسیلاتور 12MHz باشد، تایمر با فرکانس 12MHz / 12 = 1MHz می‌شمارد. یعنی در هر 1µs یک عدد می‌شمارد.
- بسته به مقدار اولیه‌ای که در THx و TLx قرار می‌گیرد، تایمر بعد از تعدادی پالس سرریز می‌دهد که این مبنای اندازه‌گیری زمان است.

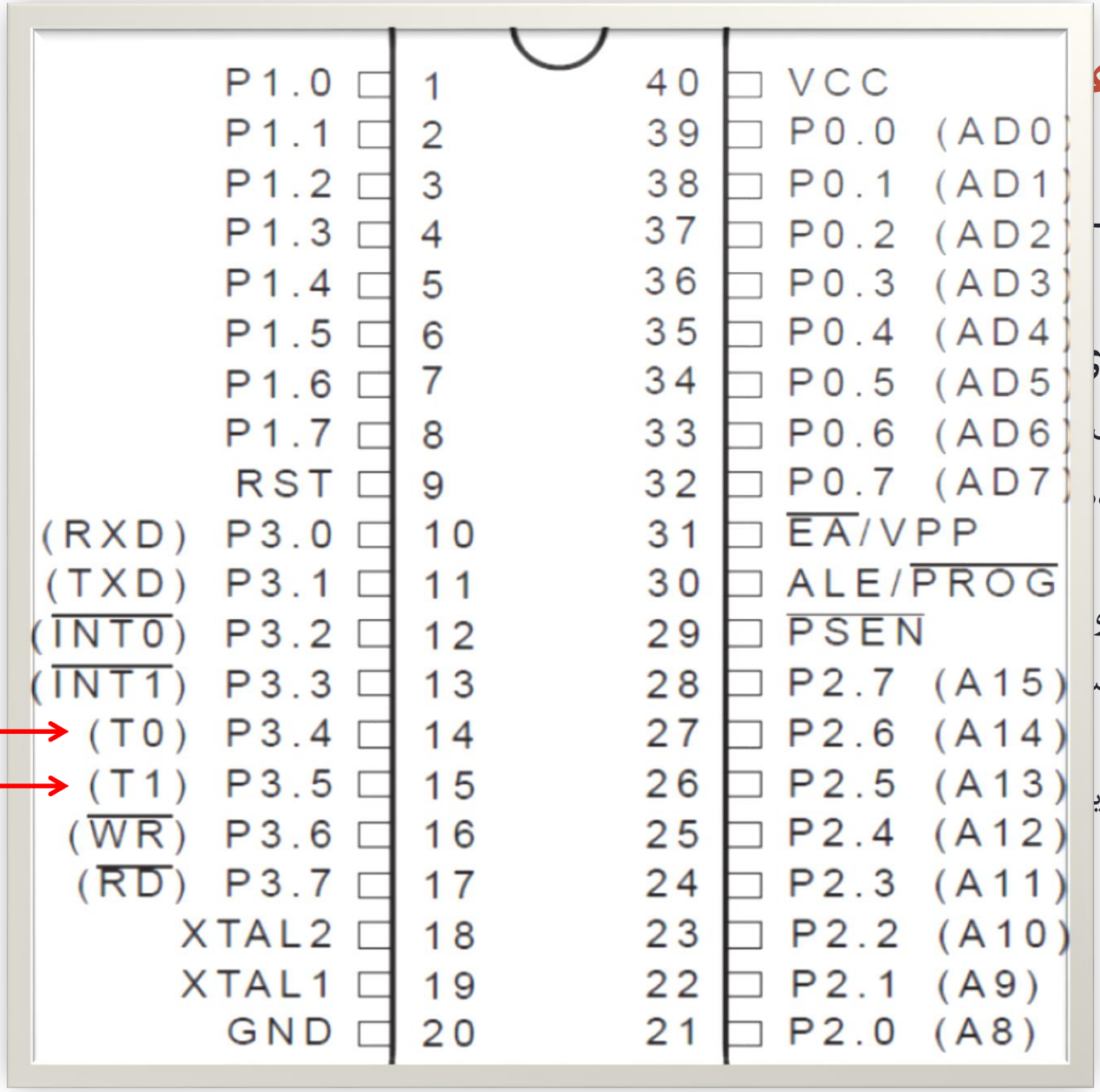


منابع پالس ساعت

- برای تایمرها دو نوع منبع پالس وجود دارد که با $C/\overline{T}$ انتخاب می‌شود.

۲. یک منبع پالس برای شمارش رویدادها

- اگر بیت $C/\overline{T}$ در ثبات TMOD یک باشد، تایمر پالس خود را از منبع خارجی تامین می‌کند.
- تایمر به عنوان شمارنده برای شمارش پالس‌های ورودی به کار می‌رود. یعنی به ازای هر پالس پایین رونده یکی می‌شمارد.
- تعداد رویدادها را می‌توان با خواندن ثبات‌های THx و TLx دانست.
- در این حالت منبع پالس خارجی در پایه T0 (P3.4) برای تایمر 0 و پایه T1 (P3.5) برای تایمر 1 اعمال می‌شود.
  - دقت شود باید ابتدا این پایه‌ها را با دستور SETB به حالت ورودی قرار داد.



منابع پالس ساع

- برای تایمرها دو نوع م

۲. یک منبع پالس برا

- اگر بیت $C/\overline{T}$ در ثبات
- تایمر به عنوان شمارند
  رونده یکی می‌شمارد.
- تعداد رویدادها را می‌تو
- در این حالت منبع پال
  اعمال می‌شود.
- دقت شود باید ابتدا ای

منبع پالس ساعت

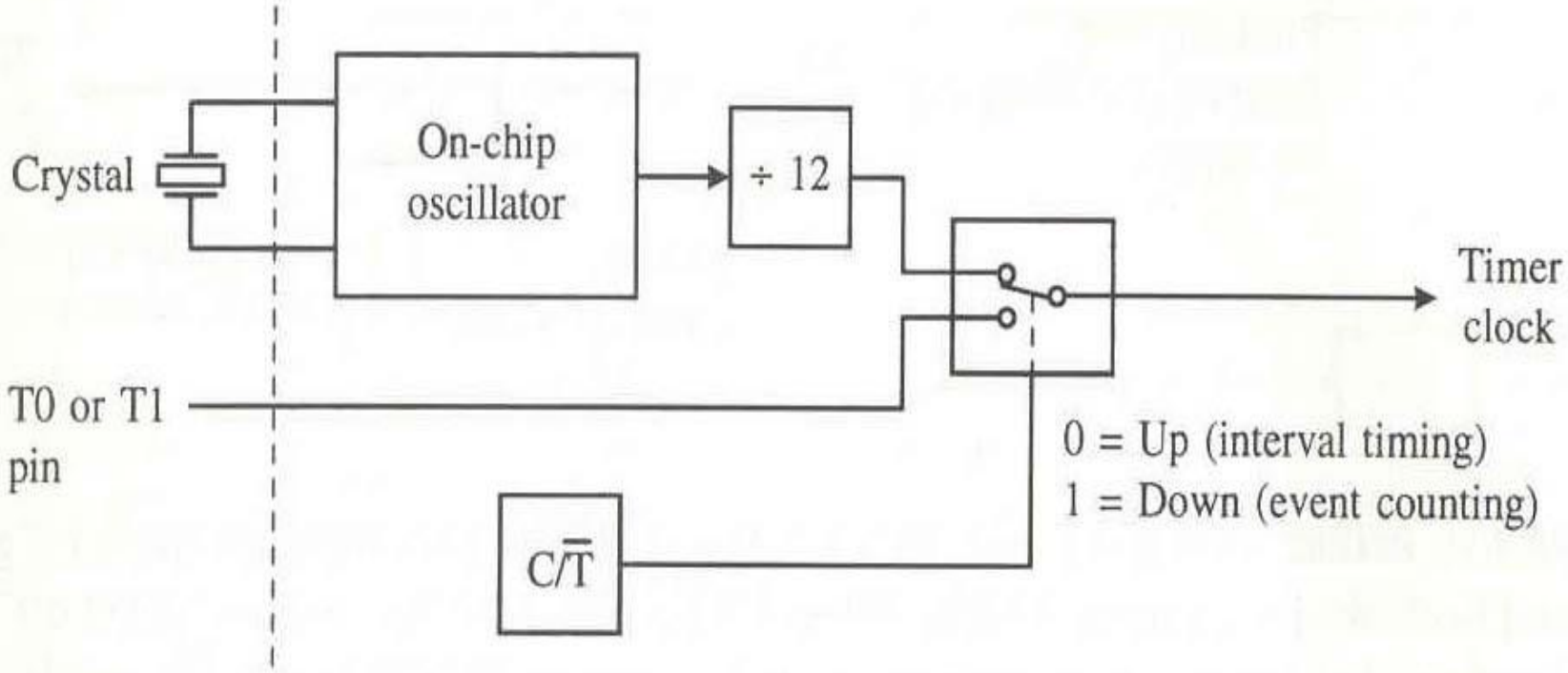
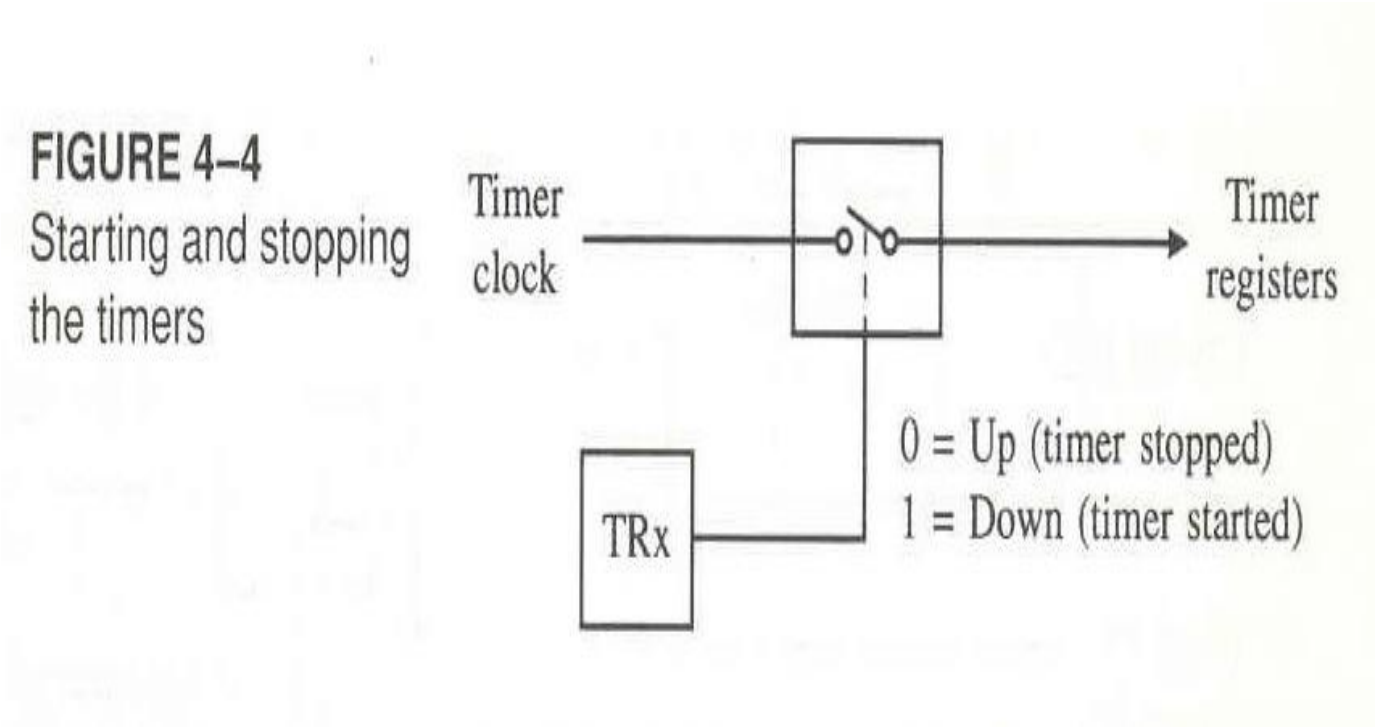


FIGURE 4–3
Clocking source

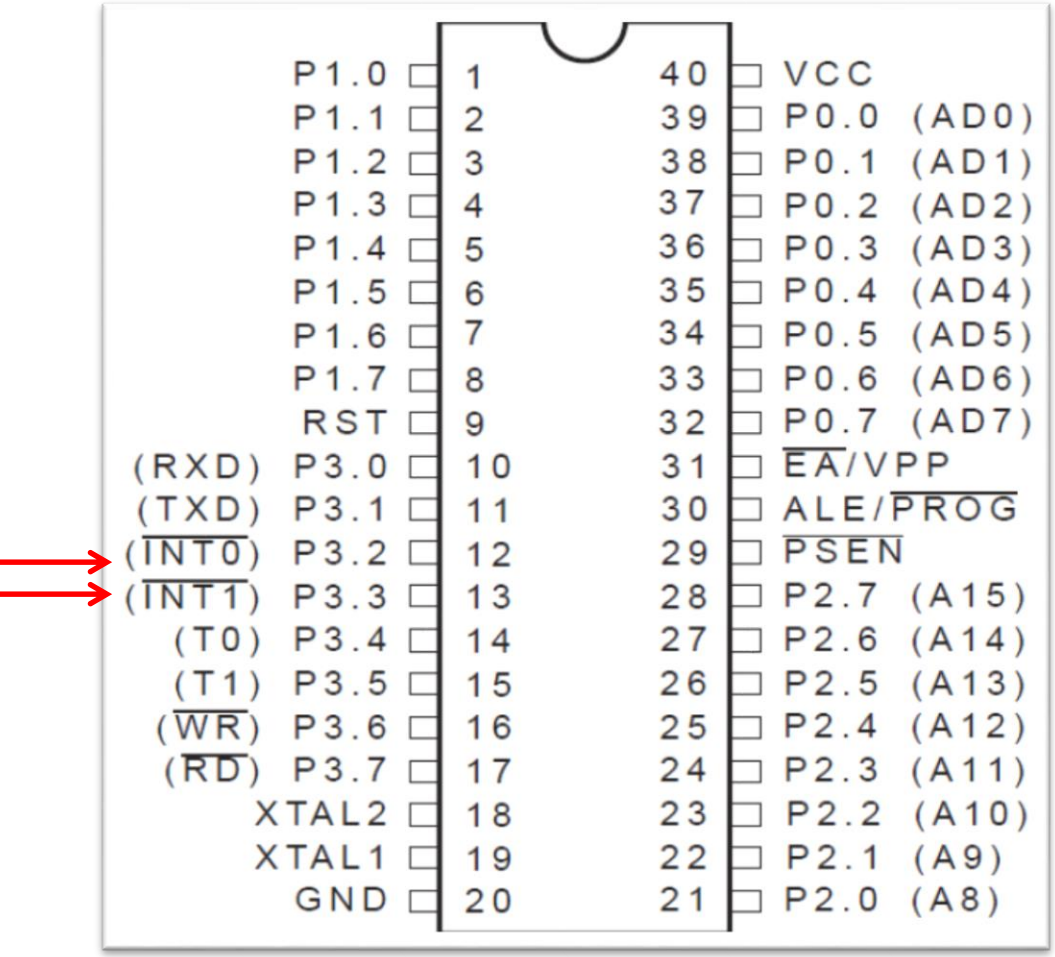
کنترل تایمرها

- یک راه برای راه‌اندازی و متوقف کردن تایمرها استفاده از TR0 و TR1 است.
  - مثال : SETB TR0 و CLR TR0



کنترل تایمرها

- روش دیگر برای کنترل تایمرها با بیت GATE در ثبات TMOD است و ورودی خارجی $\overline{INT0}$ یا $\overline{INT1}$ است.



مثال : اگر بخواهیم زمانی که پالس $\overline{INT1}$ برابر با ۱ است را اندازه بگیریم:

تایمر 1 را در مود ۱ (۱۶ بیتی) قرار می‌دهیم.

مقدار اولیه TL1=TH1=0 قرار می‌دهیم.

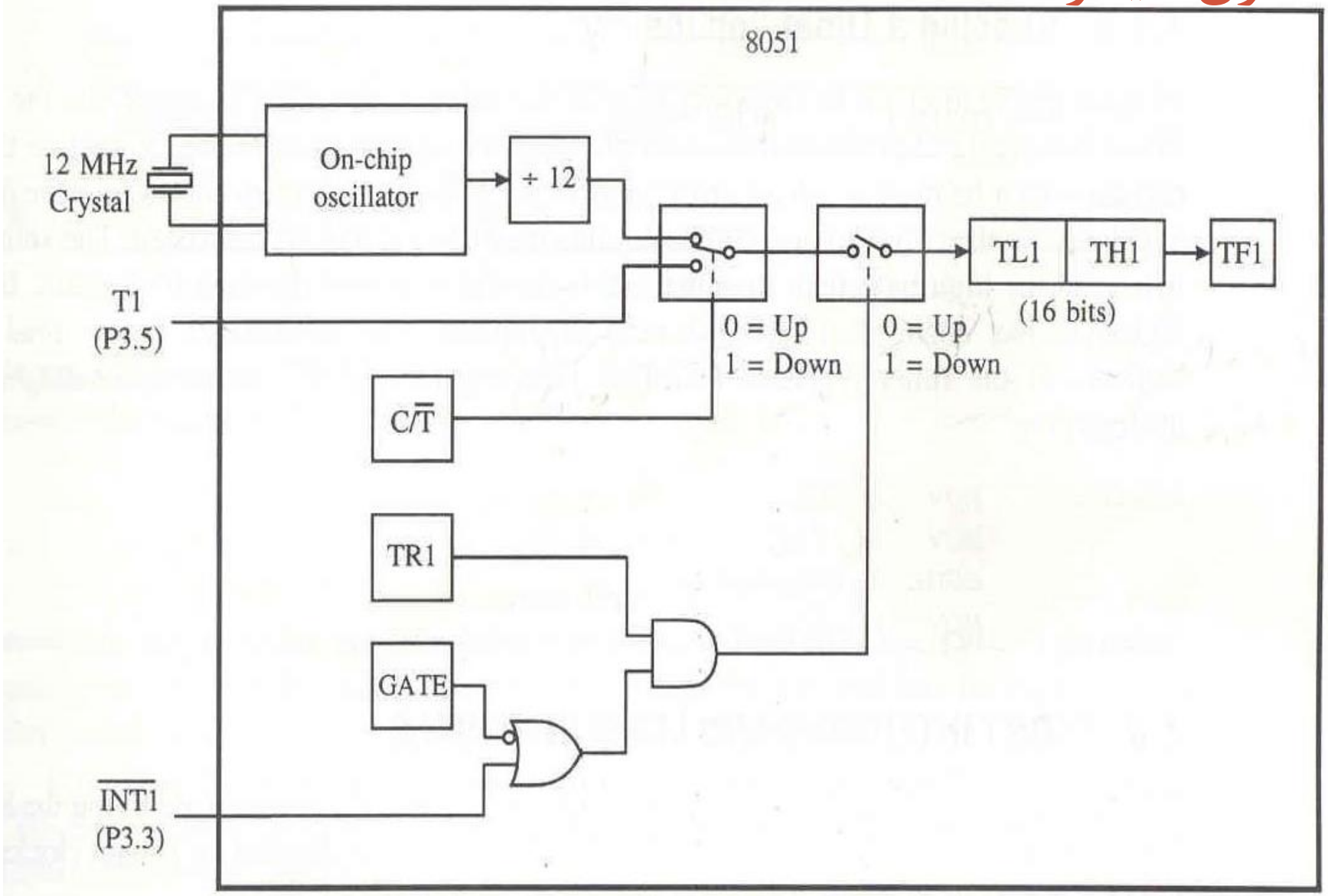
GATE=1

TR1=1

حال موقعیکه پالس $\overline{INT1}$ برابر با یک شد، تایمر با فرکانس پالس اسیلاتور شروع به شمارش می‌کند و زمانیکه $\overline{INT1}$ برابر با صفر شد متوقف می‌شود.

عددی که در TL1 و TH1 قرار دارد مدت زمان یک بودن پالس خروجی را برحسب میکروثانیه نشان می‌دهد.

کنترل تایمرها



مقدار اولیه دادن به ثبات‌های تایمر

- ثبات‌های تایمرها معمولا در ابتدای برنامه مقداردهی می‌شوند.
- TMOD اولین ثباتی است که مقدار می‌گیرد.
  - مثال : MOV TMOD, #00010000B تایمر 1 را روی مد ۱ (۱۶ بیتی) تنظیم می‌کند.
  - مثال : MOV TMOD, #00000010B تایمر 0 را روی مد ۲ (۸ بیتی با بارشدن خودکار) تنظیم می‌کند.
- تا زمانی که TRx یک نشود تایمر شروع به کار نمی‌کند.

- در صورتی که نیاز است تایمر به اندازه زمان مشخصی شمارش کند، باید به ثبات‌های TLx و THx مقدار اولیه داد.
  - مثال : اگر زمان 100µs مورد نیاز باشد :
  - در تایمر ۱۶ بیتی
    - MOV TL1, #-100 و MOV TH1, #0FFH
    - MOV TL1, #156 و MOV TH1, #0FFH
    - MOV TL1, #9CH و MOV TH1, #0FFH
  - در تایمر ۸ بیتی
    - MOV TH0, #-100
    - MOV TH0, #156
    - MOV TH0, #9CH

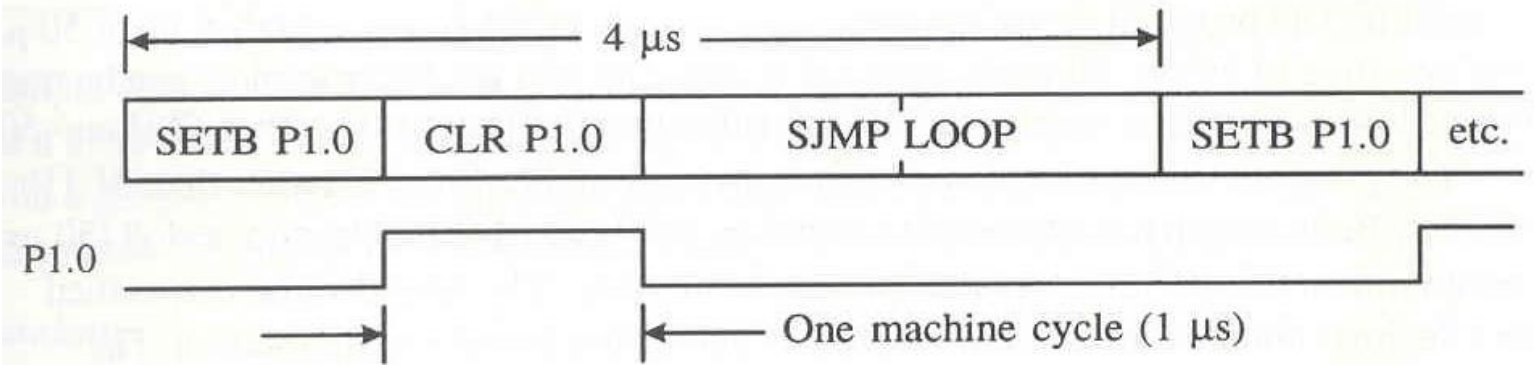
- با دستور SETB TR1 تایمر یک راه‌اندازی می‌شود و به مدت 100µs بعد TF1 یک می‌شود.
- با دستور زیر می‌توان در حلقه انتظار منتظر یک شدن TF1 ماند.
  - WAIT : JNB TF1, WAIT
- برای متوقف کردن تایمر از CLR TR1 استفاده می‌کنیم. و همچنین CLR TF1 پرچم سرریز را آماده برای استفاده‌ی مجدد می‌نماید.

ایجاد پالس

- دستورات زیر یک پالس در پین P1.0 ایجاد می‌کند. فرکانس و سیکل کاری این موج چقدر است؟

```
HERE:   SETB P1.0
        CLR P1.0
        SJMP HERE
```

- دستور ۱ پایه P1.0 را برابر با ۱ می‌کند، لذا در انتهای اجرای دستور این پایه برابر با ۱ می‌شود.
- دستور ۲ پایه P1.0 را برابر با ۰ می‌کند، لذا در انتهای اجرای دستور این پایه برابر با ۰ می‌شود.
- دستور SJMP نیاز به دوسیکل ماشین برای اجرا دارد، لذا اجرای آن 2µs طول می‌کشد.
- موجی با پریود 4µs یعنی با فرکانس 250KHz تولید می‌شود.
- این موج در مدت 1µs از 4µs برابر با ۱ است. پس سیکل کاری آن برابر است با ¼=25%

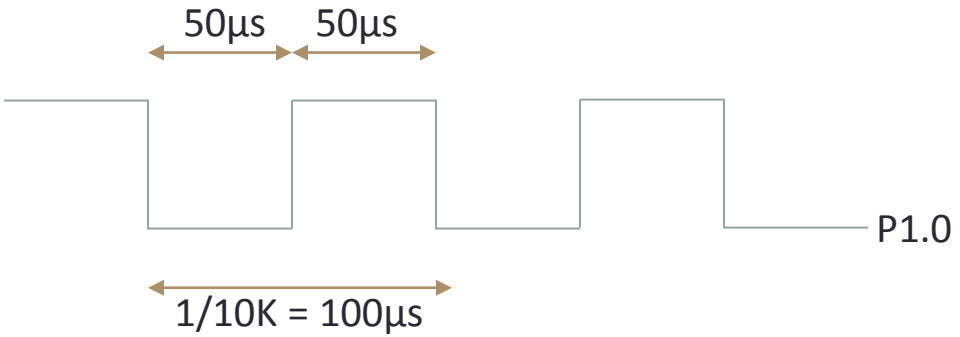


TMOD							
Timer 1				Timer 0			
7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1	M0

مثال

- برنامه‌ای بنویسید که با به کار بردن تایمر 0 موج مربعی با فرکانس 10KHz در پایه P1.0 ایجاد نماید. (فرکانس اسیلاتور 12MHz)

```
        MOV TMOD, #02H      ;timer0, 8-bit auto-reload mode
        MOV TH0, #-50
        SETB TR0
LOOP:   JNB TF0, LOOP
        CLR TF0
        CPL P1.0
        SJMP LOOP
```



موج مربعی 10KHz به مدت 50µs یک و 50µs صفر است.

برای محاسبه‌ی زمان نیاز است دو مقدار را بدانیم : فرکانس اسیلاتور و مقدار اولیه تایمر
فرکانس تایمر 1/12 فرکانس اسیلاتور است. در اینجا فرکانس تایمر 12MHz/12=1MHz
یعنی تایمر در هر 1/1MHz=1µs یک واحد شمارش می‌کند.
ما نیاز داریم زمان‌های 50µs را تولید کنیم. پس کافی است مقدار اولیه تایمر را 50- تعیین کنیم.
چون 50µs کمتر از 256 است پس می‌توان آن را با تایمر ۸ بیتی شمارش نمود.
برای مقداردهی تایمر برای شمارش 50µs می‌توان از سه روش زیر استفاده کرد.

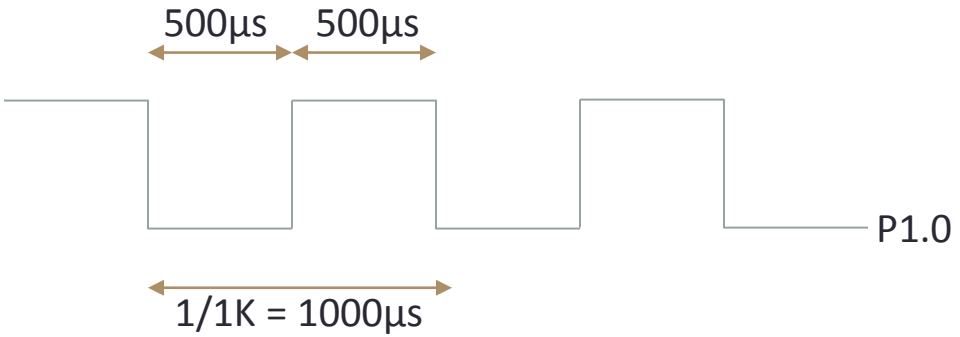
TH0=Hex(-50)=Hex(206)=CEH TH0=256-50=206 TH0=-50

TMOD							
Timer 1				Timer 0			
7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1	M0

مثال

- برنامه‌ای بنویسید که با به کار بردن تایمر 0 موج مربعی با فرکانس 1KHz در پایه P1.0 ایجاد نماید. (فرکانس اسیلاتور 12MHz)

```
        MOV TMOD, #01H      ;timer0, 16-bit mode
LOOP:   MOV TH0, #0FEH
        MOV TL0, #0CH
        SETB TR0
WAIT:   JNB TF0, WAIT
        CLR TR0
        CLR TF0
        CPL P1.0
        SJMP LOOP
```



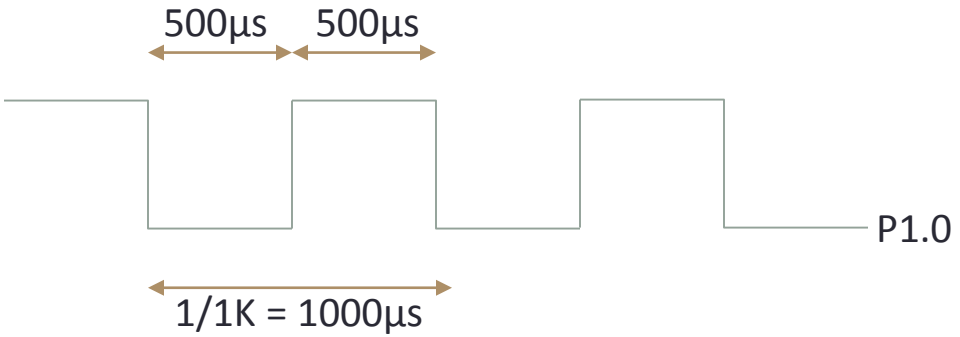
موج مربعی 1KHz به مدت 500µs یک و 500µs صفر است.
شمارش 500µs توسط تایمر ۸ بیتی امکان پذیر نیست (500>256)
پس از تایمر ۱۶ بیتی استفاده می‌کنیم.
برای مقداردهی اولیه ثبات های TH0 و TH1 باید مقدار را ابتدا به هگزادسیمال تبدیل کرد و سپس ۸ بیت پرارزشتر را در TH0 و ۸ بیت کم ارزشتر را در TL0 ریخت.
-500 = 65536-500 = FE0CH

TMOD							
Timer 1				Timer 0			
7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1	M0

مثال

- برنامه‌ای بنویسید که با به کار بردن تایمر 0 موج مربعی با فرکانس 1KHz در پایه P1.0 ایجاد نماید. (فرکانس اسیلاتور 12MHz)

```
        MOV TMOD, #01H      ;timer0, 16-bit mode
LOOP:   MOV TH0, #0FEH
        MOV TL0, #0CH
        SETB TR0
WAIT:   JNB TF0, WAIT
        CLR TR0
        CLR TF0
        CPL P1.0
        SJMP LOOP
```



دقت شود در تایمر ۱۶ بیتی پس از هر بار رخ دادن سرریز باید تایمر دوباره مقداردهی شود و گرنه شمارش از صفر تا ۶۵۵۳۶ خواهد بود.

در تایمر ۸ بیتی مقداردهی تایمر بعد از هر سرریز به صورت خودکار صورت می‌گیرد.

در اینجا فرکانس موج مربعی اندکی با 1KHz تفاوت دارد. که به علت زمانی است که صرف مقدار اولیه دادن به تایمر می‌شود.

می‌توان مقدار اولیه‌ی تایمر را طوری تنظیم کرد که فرکانس موج دقیقا برابر با 1KHz شود. این مقادیر را پیدا کنید.

در حالت مود ۲ (بار شدن اتوماتیک) این خطا هیچگاه رخ نمی‌دهد. اما به علت ۸ بیتی بودن تایمر تنها می‌توان برای تولید فرکانس‌های بالا از آن استفاده نمود.

محاسبه زمان شمارش شده توسط تایمر ۱۶ بیتی

1. محاسبه تعداد پالس‌های تایمر تا زمان سرریز
   - در حالت هگزادسیمال
     - تعداد پالس‌های تایمر : FFFF – YYXX + 1
     - که در آن مقدار XX = TLx و YY = THx است بر حسب هگزادسیما است.
   - در حالت دهدهی
     - تعداد پالس‌های تایمر : 65535 – NNNN + 1
     - Dec(YYXX) = NNNN
     - که در آن مقدار YYXX (XX = TLx و YY = THx) است که معادل دهدهی آن NNNN است.

2. محاسبه زمان هر پالس تایمر
   - فرکانس تایمر = ۱۲ / فرکانس اسیلاتور
   - زمان هر پالس = فرکانس تایمر / ۱

3. زمان شمارش شده توسط تایمر = زمان هر پالس تایمر × تعداد پالس‌های تایمر

محاسبه زمان شمارش شده توسط تایمر ۸ بیتی

1. محاسبه تعداد پالس‌های تایمر تا زمان سرریز
   - در حالت هگزادسیمال
     - تعداد پالس‌های تایمر : FF – YY + 1
     - که در آن مقدار YY = THx است بر حسب هگزادسیما است.
   - در حالت دهدهی
     - تعداد پالس‌های تایمر : 255 – NN + 1
     - Dec(YY) = NN
     - که در آن مقدار YY (YY = THx) است که معادل دهدهی آن NN است.

2. محاسبه زمان هر پالس تایمر
   - فرکانس تایمر = ۱۲ / فرکانس اسیلاتور
   - زمان هر پالس = فرکانس تایمر / ۱

3. زمان شمارش شده توسط تایمر = زمان هر پالس تایمر × تعداد پالس‌های تایمر

محاسبه مقدار اولیه تایمر ۱۶ بیتی

1. محاسبه زمان هر پالس تایمر
   - فرکانس تایمر = ۱۲ / فرکانس اسیلاتور
   - زمان هر پالس = فرکانس تایمر / ۱

- محاسبه تعداد پالس‌های مورد نیاز برای ایجاد تاخیر زمانی در نظر گرفته شده
  - A = تعداد پالس‌ها = زمان هر پالس / تاخیر زمانی

- محاسبه مقدار اولیه تایمر
  - Hex(-A) = Hex(65535-A+1) = YYXXH
  - THx = YYH TLx = XXH

محاسبه مقدار اولیه تایمر ۸ بیتی

1. محاسبه زمان هر پالس تایمر
   - فرکانس تایمر = ۱۲ / فرکانس اسیلاتور
   - زمان هر پالس = فرکانس تایمر / ۱

- محاسبه تعداد پالس‌های مورد نیاز برای ایجاد تاخیر زمانی در نظر گرفته شده
  - A = تعداد پالس‌ها = زمان هر پالس / تاخیر زمانی

- محاسبه مقدار اولیه تایمر
  - Hex(-A) = Hex(255-A+1) = YYH
  - THx = YYH

TMOD							
Timer 1				Timer 0			
7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1	M0

مثال

- در برنامه‌ی زیر، موج مربعی با فرکانس کاری ۵۰٪ (نصف یک و نصف صفر) ایجاد شده است. فرکانس این موج تقریبا چقدر است؟ (فرکانس اسیلاتور 12MHz)

```
        MOV   TMOD,#01    ;Timer 0, mode 1(16-bit mode)
HERE:   MOV   TL0,#0F2H   ;TL0=F2H, the low byte
        MOV   TH0,#0FFH   ;TH0=FFH, the high byte
        CPL   P1.5        ;toggle P1.5
        ACALL DELAY
        SJMP HERE
DELAY:
        SETB TR0          ;start the timer 0
AGAIN:  JNB   TF0,AGAIN   ;monitor timer flag 0
                          ;until it rolls over
        CLR   TR0         ;stop timer 0
        CLR   TF0         ;clear timer 0 flag
        RET
```

FFFF - FFF2 = D = 13

13+1=14

یا

Dec(FFF2)=65522

65536-65522=14

فرکانس اسیلاتور 12MHz است پس تایمر در هر 1µs یک واحد می‌شمارد.

پس پالس در 14µs یک و در 14µs صفر است.

پس فرکانس این موج مربعی برابر است با 1/(2*14µ)= 35KHz

مثال

- در مثال قبل، فرکانس موج را بصورت دقیق محاسبه کنید.

```
                          Cycles
HERE:  MOV  TL0,#0F2H       2
       MOV  TH0,#0FFH       2
       CPL  P1.5            1
       ACALL DELAY          2
       SJMP  HERE           2
DELAY:
       SETB TR0             1
AGAIN: JNB  TF0,AGAIN      14
       CLR  TR0             1
       CLR  TF0             1
       RET                  2
                 Total      28
```

دستورات نیز در این برنامه دارای تاخیر هستند و این باعث می‌گردد مدت زمانی که پالس صفر و یک است دقیقا برابر با 14µs نباشد.

به طور دقیق به اندازه 28 سیکل ماشین یعنی 28µs پالس صفر و به اندازه ی 28 سیکل ماشین یعنی 28µs پالس یک است. پس فرکانس این موج مربعی برابر است با 1/(2*28µ)= 17KHz

مثال

- با استفاده از هر دو روش مطرح شده در اسلاید قبل، زمان یک بودن پالس تولید شده در برنامه زیر را محاسبه کنید. سربار دستورات اضافی را در نظر نگیرید. (فرکانس اسیلاتور 12MHz)

```
        CLR   P2.3        ;Clear P2.3
        MOV   TMOD,#01    ;Timer 0, 16-bitmode
HERE:   MOV   TL0,#3EH    ;TL0=3Eh, the low byte
        MOV   TH0,#0B8H   ;TH0=B8H, the high byte
        SETB  P2.3        ;SET high timer 0
        SETB  TR0         ;Start the timer 0
AGAIN:  JNB   TF0,AGAIN   ;Monitor timer flag 0
        CLR   TR0         ;Stop the timer 0
        CLR   TF0         ;Clear TF0 for next round
        CLR   P2.3
```

1)

a) (FFFF-B83E+1) = 47C2 H = 18370 in decimal

b) B83E = 47166 in decimal

65535-47166+1 = 18370

2) 12MHz / 12 = 1MHz

1 / 1MHz = 1µs

3) 18370 * 1µs = 18370µs

مثال

- اگر فرکانس اسیلاتور XTAL=11.0592 MHz باشد، و بخواهیم در برنامه‌ی زیر یک پالس با طول 5ms در P2.3 ایجاد کنیم، مقدار اولیه تایمر یک چه باید باشد؟

11.0592 M / 12 = 921600 Hz

1 / 921600 = 1.085 µs

یعنی تایمر در هر 1.085µs یک واحد می‌شمارد.

5ms / 1.085µs = 4608 clocks

65535 – 4608 + 1 = 60928

Hex(60928) = EE00 H

TL = 00H , TH = EEH

```
        CLR  P2.3      ;Clear P2.3
        MOV  TMOD,#01  ;Timer 0, 16-bitmode
HERE:   MOV  TL0,#0    ;TL0=0, the low byte
        MOV  TH0,#0EEH ;TH0=EE, the high byte
        SETB P2.3      ;SET high P2.3
        SETB TR0       ;Start timer 0
AGAIN:  JNB  TF0,AGAIN ;Monitor timer flag 0
        CLR  TR0       ;Stop the timer 0
        CLR  TF0       ;Clear timer 0 flag
        CLR  P2.3
```

مثال

- اگر فرکانس اسیلاتور XTAL=11.0592 MHz باشد، برنامه‌ای بنویسید که موج مربعی با فرکانس 50Hz بر روی P2.3 ایجاد کند.

Solution:

Look at the following steps.

(a) T = 1 / 50 = 20 ms, the period of square wave.

(b) 1 / 2 of it for the high and low portion of the pulse is 10 ms.

(c) 10 ms / 1.085 us = 9216 and 65536 – 9216 = 56320 in decimal, and in hex it is DC00H.

(d) TL = 00 and TH = DC (hex).

```
        MOV  TMOD,#10H  ;Timer 1, mod 1
AGAIN:  MOV  TL1,#00    ;TL1=00,low byte of timer
        MOV  TH1,#0DCH ;TH1=DC, the high byte
        SETB TR1        ;Start timer 1
BACK:   JNB  TF1,BACK   ;until timer rolls over
        CLR  TR1        ;Stop the timer 1
        CPL  P2.3       ;Comp. p2.3 to get hi, lo
        SJMP AGAIN      ;Reload timer
                        ;mode 1 isn't auto-reload
```

مثال

- میزان تاخیر ایجاد شده در برنامه زیر را با به صورت تقریبی محاسبه کنید.

```
        MOV   TMOD,#10H  ;Timer 1, mod 1
        MOV   R3,#200    ;cnter for multiple delay
AGAIN:  MOV   TL1,#08H   ;TL1=08,low byte of timer
        MOV   TH1,#01H   ;TH1=01,high byte
        SETB  TR1        ;Start timer 1
BACK:   JNB   TF1,BACK   ;until timer rolls over
        CLR   TR1        ;Stop the timer 1
        CLR   TF1        ;clear Timer 1 flag
        DJNZ  R3,AGAIN   ;if R3 not zero then
                         ;reload timer
```

Solution:

TH-TL = 0108H = 264 in decimal and 65536 – 264 = 65272. Now
65272 × 1.085 μs = 70.820 ms, and for 200 of them we have
200 ×70.820 ms = 14.164024 seconds.

TMOD							
Timer 1				Timer 0			
7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1	M0

مثال

- فرکانس موج مربعی ایجاد شده را محاسبه نمایید.

```
        MOV   TMOD,#2H   ;Timer 0, mod 2
                         ;(8-bit, auto reload)
        MOV   TH0,#0
AGAIN:  MOV   R5,#250     ;multiple delay count
        ACALL DELAY
        CPL   P1.0
        SJMP  AGAIN

DELAY:  SETB  TR0        ;start the timer 0
BACK:   JNB   TF0,BACK   ;stay timer rolls over
        CLR   TR0        ;stop timer
        CLR   TF0        ;clear TF for next round
        DJNZ  R5,DELAY
        RET
```

$$T = 2\,(\,250 \times 256 \times 1.085\ \text{us}\,) = 138.88\text{ms, and frequency} = 72\ \text{Hz}$$

مثال

- فرض کنید می‌خواهیم از تایمر 1 در مود ۲ استفاده کنیم. برای ایجاد تاخیرهای زیر، چه مقدار اولیه‌ای باید به TH1 داده شود؟
- 200µs , 60 µs , 3 µs , 12 µs , 48 µs

- راه حل : استفاده از ماشین حساب ویندوز در مود برنامه‌نویسی
- MOV TH1, #-200 یا MOV TH1, #38H
- MOV TH1, #-60 یا MOV TH1, #0C4H
- MOV TH1, #-3 یا MOV TH1, #0FDH
- MOV TH1, #-12 یا MOV TH1, #0F4H
- MOV TH1, #-48 یا MOV TH1, #0D0H

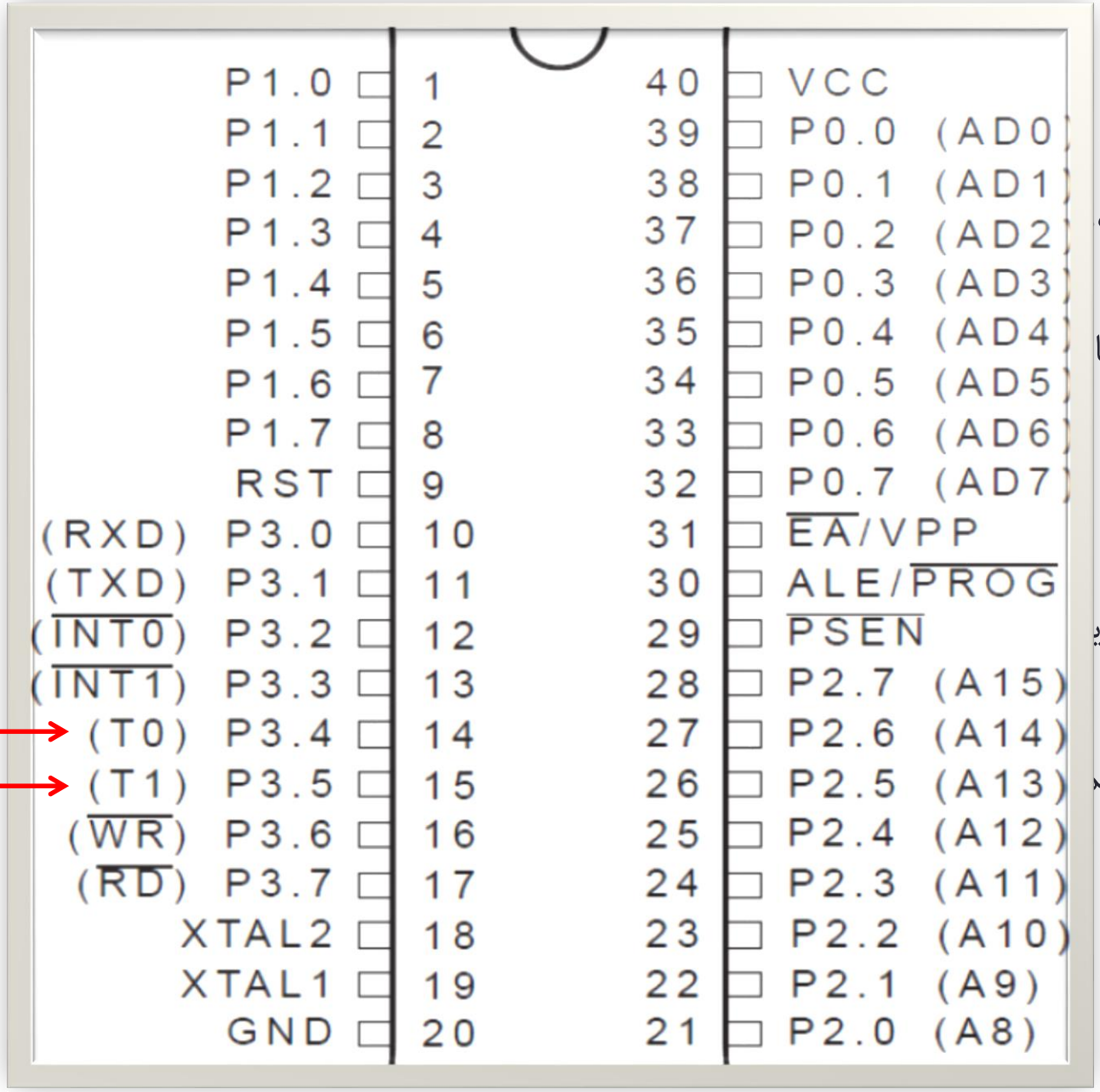
شمارنده

- همانطور که گفته شد، تایمر می‌تواند به عنوان شمارنده برای شمارش اتفاقات خارجی استفاده شود.
  - در این حالت، پالس‌هایی که بیرون از میکروکنترلر اتفاق می‌افتند و به پایه‌ی میکروکنترلر متصل هستند، باعث شمارش در تایمر می‌شوند.

- مقادیر TMOD، TH و TL همانند تایمرها تنظیم می‌شود.
  - باید بیت C/T برابر با یک باشد تا تایمر به عنوان شمارنده کار کند.

- تنها منبع پالس در شمارنده با تایمر متفاوت است.
  - پالس خارجی از طریق پین T0 برای تایمر 0 و از طریق پین T1 برای تایمر 1 تامین می‌گردد.

شمارنده



- همانطور که گفته شد استفاده شود.
  - در این حالت، پالس ها هستند، باعث شمارش
- مقادیر TH ،TMOD
  - باید بیت C/T برابر با ی
- تنها منبع پالس در ش
  - پالس خارجی از طریق

TMOD							
Timer 1				Timer 0			
7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1	M0

مثال

- فرض کنید پالسی از طریق پین T1 به میکروکنترلر وارد می‌شود. همچنین فرض کنید یک LCD به پورت P2 متصل است. با استفاده از تایمر برنامه‌ای بنویسید که تعداد پالس‌های رخ داده را در LCD نمایش دهد.

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