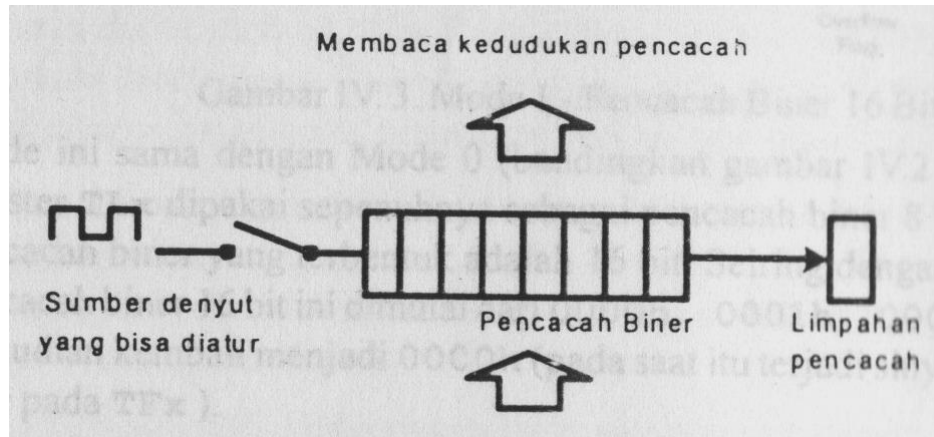


Timer dan Counter

Konsep dasar timer/counter



Sinyal clock dibedakan menjadi dua yaitu :

1. Clock dengan frekuensi tetap ---> pencacah ini disebut timer
2. Clock dengan frekuensi variabel ----> pencacah ini disebut counter

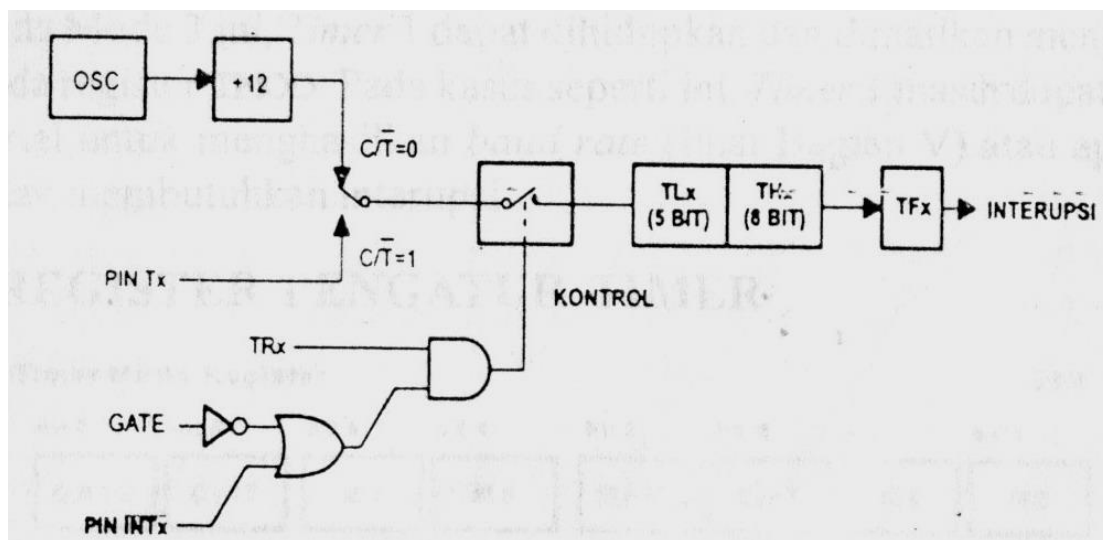
AT89C51 dan AT89Cx051 mempunyai timer 0 dan timer 1, sedangkan untuk AT89C52/55 mempunyai timer 2.

Untuk mengakses timer digunakan SFR. Timer 0 diakses melalui register TL0 dan TH0, timer 1 melalui register TL1 dan TH1.

AT89C51/52 merupakan pencacah biner 16 bit yang mencacah dari 0000h sampai FFFFh, saat kondisi pencacah berubah dari FFFFh kembali ke 0000h akan timbul sinyal overflow.

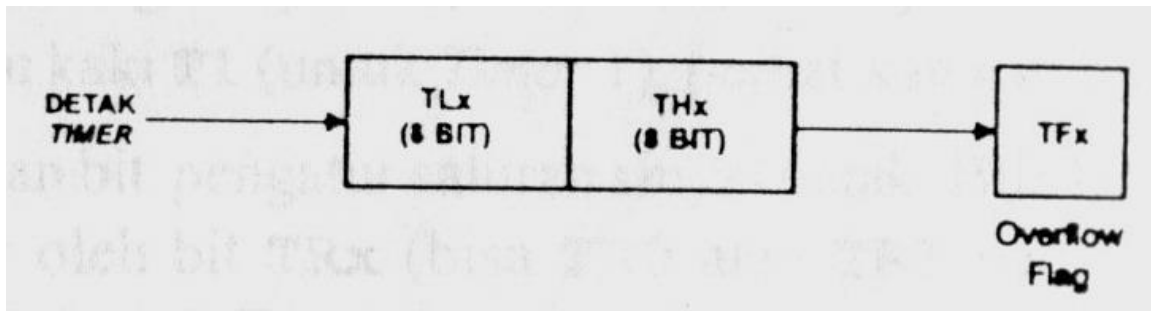
Untuk mengatur kerja timer/counter digunakan register TCON dan TMOD

Mode 0 – Pencacah biner 13 bit



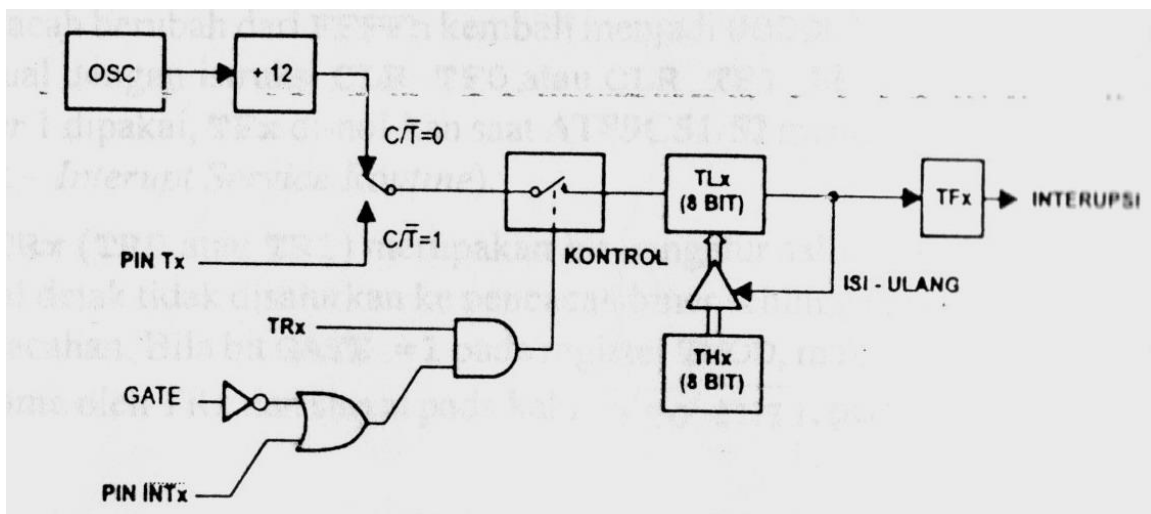
Dibentuk dengan TLx sebagai pencacah biner 5 bit, limbahnya dihubungkan ke THx. Limpaan pencacah 13 bit ini ditampung di TFx yang berada di TCON. Bila terjadi overflow maka TFx akan diset (=1) Masukan ke pencacah (baik dari eksternal (Tx) maupun internal ($1/12 F_{osc}$)) diaktifkan jika $TRx=1$ dan $gate=0$ atau $\overline{INTx}=1$, maka keluaran dari gerbang OR menjadi selalu 1 dan akibatnya hasil gerbang AND juga 1. Jika $gate=1$ maka timer sepenuhnya dikendalikan oleh masukan eksternal \overline{INTx} dan bisa digunakan dalam pengukuran lebar pulsa. TRx merupakan bit kontrol dalam TCON.

Mode 1-Pencacah biner 16 bit



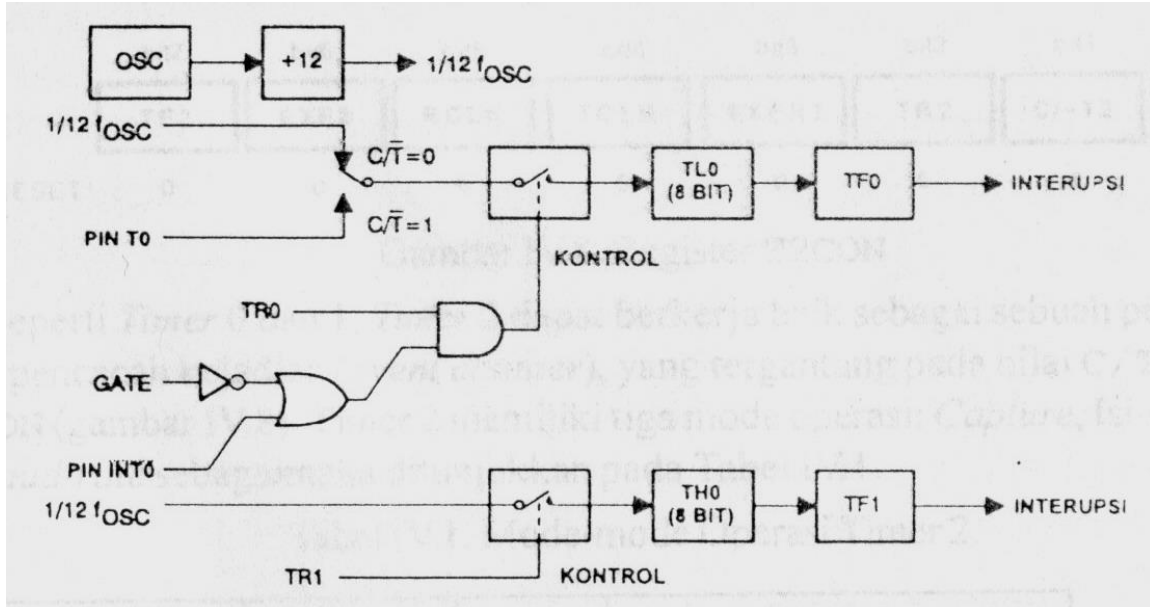
Mirip dengan mode 0, tetapi register TLx dipakai 8 bit

Mode 2-Pencacah biner 8 bit dengan isi ulang



TLx digunakan sebagai pencacah biner 8 bit, sedangkan THx digunakan untuk menyimpan nilai yang diisi ulang ke TLx setiap kali kondisi TLx overflow. Dengan cara ini dapat diperoleh sinyal overflow yang frekuensinya dapat ditentukan dari nilai yang disimpan dalam THx

Mode 3-Gabungan pencacah biner 16 bit dan 8 bit



TL0, TH0, TL1, dan TH1 digunakan untuk membentuk 3 rangkai pencacah : pencacah biner 16 bit (oleh TL1 dan TH1), pencacah biner 8 bit (oleh TL0 dan TF0 sebagai sarana pemantau overflow, dan pencacah biner 8 bit (oleh TH0 dan TF1 sebagai sarana pemantau overflow. TH0 mengendalikan interupsi timer 1 (TF1). Timer 1 dinyalakan atau dimatikan dengan menggunakan M1 dan M0 pada TMOD.

Register Pengatur Timer

TMOD-Timer Mode Register								89H
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Gate	C/-T	M1	M0	Gate	C/-T	M1	M0
Reset	0	0	0	0	0	0	0	0
	Timer 1				Timer 2			

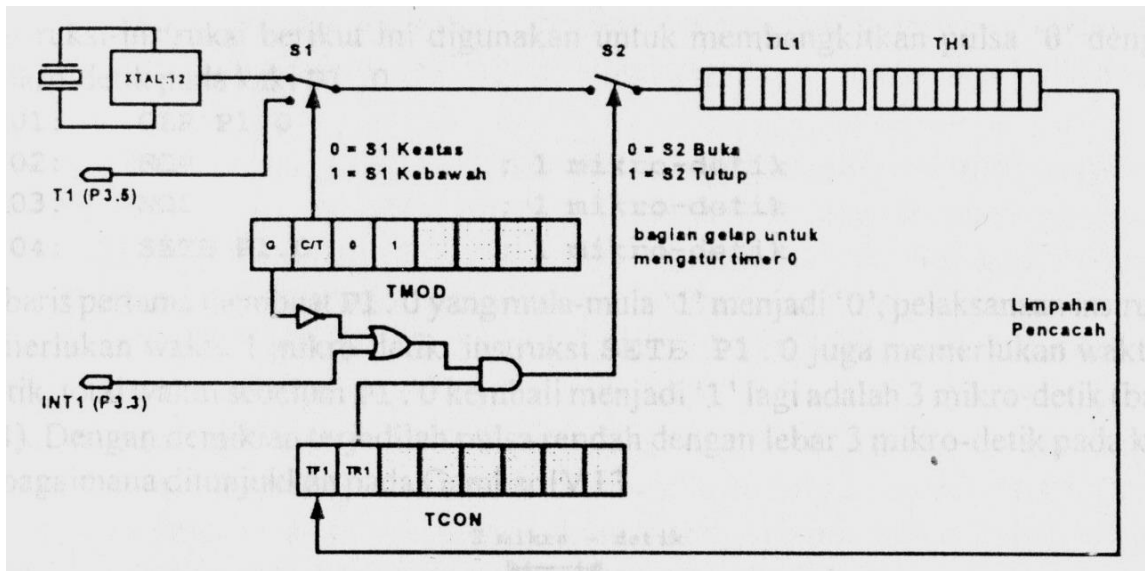
M1	M0	Mode
0	0	0
0	1	1
1	0	2
1	1	3

- M0/M1 untuk menentukan mode
- C/-T untuk mengatur sumber sinyal detak yang diberikan ke pencacah biner. Jika 0 maka sinyal dari osilator kristal yang frekuensinya sudah dibagi 12, jika 1 maka sinyal dari kaki T0 (untuk timer 0) atau kaki T1 (untuk timer 1)
- Gate pengatur saluran clock. Bila gate=0, saluran hanya diatur oleh bit TRx (TR0 atau TR1 pada TCON). Bila gate=1 kaki INT0 atau INT1 yang digunakan untuk mengatur saluran sinyal clock

TCON-Timer Control Register								88H
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Reset	0	0	0	0	0	0	0	0
	Timer 1		Timer 0		INT1 dan INT0			

- TFx (TF0 atau TF1) merupakan bit penampung limpahan, TFx akan '1' setiap kali pencacah biner over flow. Bit ini dapat di nol kan secara manual.
- TRx (TR0 atau TR1) merupakan bit pengatur saluran clock. Bila '0' clock tidak disalurkan ke pencacah biner sehingga pencacah berhenti melakukan pencacahan. Bila gate=1 pada TMOD, sinyal clock diatur bersama TRx dan sinyal kaki INT0/INT1.

Mengatur Timer



Rangkaian timer yang dikehendaki dibentuk dengan register TMOD, sedangkan kerja timer dikendalikan oleh register TCON. Setelah mikrokontroler direset, register TMOD bernilai 00h, berarti :

- Bila C/-T=0, menurut gambar, keadaan ini membuat saklar S1 ke posisi atas, sumber clock berasal dari osilator kristal yang frekuensinya sudah dibagi 12, pencacah biner yang dibentuk dengan TL1 dan TH1 berfungsi sebagai timer. Jika sistem dirancang untuk timer 1 bekerja sebagai timer maka C/-T tidak perlu diubah. Tetapi bila sebagai counter untuk menghitung pulsa yang masuk lewat kaki T1 (p3.5), maka posisi saklar S1 harus kebawah dengan membuat C/-T menjadi '1'.
- Jika gate='0'. Membuat gerbang OR selalu '1' tidak dipengaruhi keadaan '0' atau '1' pada kaki INT1 (P3.3). Dalam keadaan ini saklar S2 hanya dikendalikan lewat bit TR1 dalam register TR. Jika TR1='1' saklar S2 tertutup sehingga clock dari S1 disalurkan ke pencacah biner. Clock akan berhenti jika TR1='0'

- Jika $gate = '1'$. Output gerbang OR akan mengikuti keadaan kaki INT1, saat $INT1 = 0$ apapun keadaan bit TR1 output AND selalu '0' dan saklar S1 selalu terbuka, agar saklar S1 dapat tertutup kaki INT1 dan bit TR1 harus = '1' secara bersamaan.
- Bit M1 dan $M0 = '0'$, berarti TL1 dan TH1 disusun menjadi pencacah biner 13 bit (mode 0), jika dikehendaki timer 1 bekerja pada mode 1 maka M1 dibuat '0' dan bit M0 '1'

TMOD=00h berarti timer 1 bekerja sebagai pencacah biner 13 bit, sumber clock dari osilator

Contoh program dengan menggunakan timer 1 sebagai delay :

```
Mov TMOD,#00100000b    ; Timer 1 mode 2 (8 bit autoreload)
Mov TH1,#F6H            ; Nilai autoreload TL1 (dapat diisi -10)
Setb TR1                ; Timer 1 mulai bekerja
```

Ulang:

```
Jnb TF1,$              ;Tunggu sampai terjadi overflow
Cpl p1.0                ; Keadaan pada p1.0 dibalik
Clr TF1                 ; Hapus overflow pencacah
Sjmp Ulang
```

Delay yang dihasilkan adalah 10 μ detik. Frekuensi sinyal dari P1.0 sebesar 1/10 mikrodetik=100kHz. Nilai awal yang diberikan ke pencacah biner diatas adalah nilai negatif dari faktor pembagi yang dikehendaki.

Contoh membuat delay 50 ms :

Start:

```
Mov TMOD,#00010000b    ; Timer 1 mode 1
Mov TL1,#LOW (-50000)   ; Siapkan delay 50 ms
Mov TH1,#High (-50000)
Clr TF1
Setb TR1
Jnb TF1,$
Clr TR1
Ret
```

Jika osilator yang digunakan 12 Mhz, delay yang dihasilkan adalah $50.000 \times 1 \mu$ detik

Contoh timer digunakan sebagai counter :

```
Mov tmod,#00001110b    ;mode 8 bit autoreload
Ulang: mov TH0,#-5
      Mov TL0,#-5
      Setb TR0          ;aktifkan timer 0
Cek:  jnb TF0,$
      Mov P1,#01h       ;Hidupkan p1.0 setelah 5 kali clock T0 (P1.4)
      Clr TF0           ; Clear overflow
Cek1: jnb TF0,$         ;Tunggu hingga overflow
```

Timer dan Counter Mikrokontroler
Oleh Ronny Mardiyanto, MT

Mov p1,#00h
Clr TF0
Sjmp Cek

; Matikan P1 setelah 5 kali clock T0
;Clear overflow