

8051

**SEL433 – Aplicação de
Microprocessadores I**

Aula 5

Prof. Adilson Gonzaga

Temporizadores e Contadores (Timer/Counter)

- O 8051 possui 2 T/C internos de 16 Bits programáveis e com capacidade de operação independente da CPU .
- Contadores crescentes (up-counter) que geram sinal de interrupção na ocorrência de overflow.
- Podem ser habilitados ou desabilitados por Software ou por Hardware.

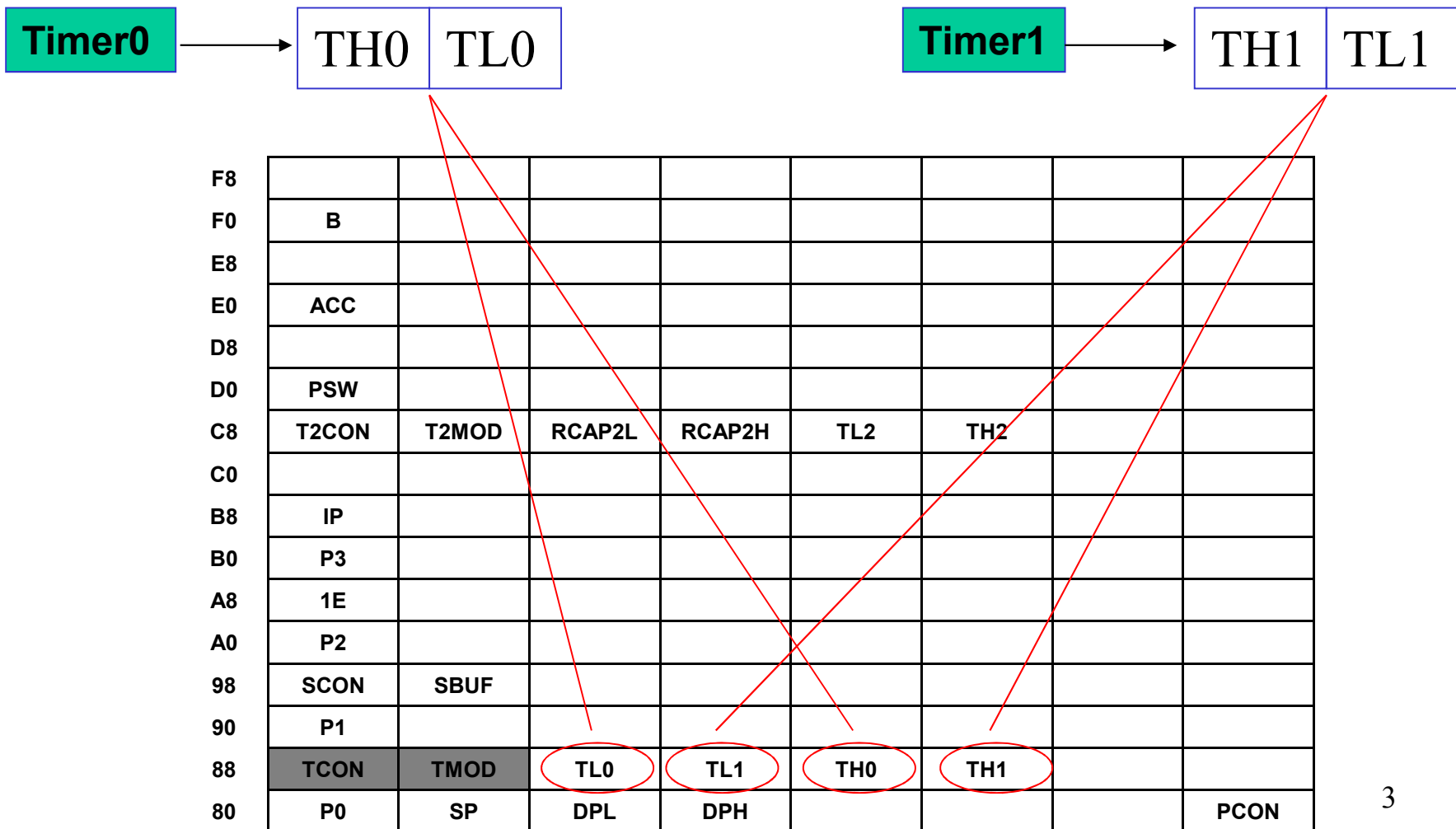
Dois registradores (SFR) comandam a programação dos T/C

•TCON

•TMOD

Temporizadores e Contadores (Timer/Counter)

Registradores na área dos SFR que mostram os valores dos T/C



Temporizadores e Contadores (Timer/Counter)

Registrador TCON

(Timer/Counter Control) → **Endereçável a Bit**

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TF1	TCON.7	Flag de overflow do Timer 1. Ativado por hardware quando o Timer 1 transborda. Zerado por hardware assim que o processador salta para a rotina de atendimento da interrupção					
TR1	TCON.6	Bit de controle do Timer 1. Ativado/zerado por software para Ligar/Desligar o Timer 1.					
TF0	TCON.5	Flag de overflow do Timer 0. Ativado por hardware quando o Timer 0 transborda. Zerado por hardware assim que o processador salta para a rotina de atendimento da interrupção					
TR0	TCON.4	Bit de controle do Timer 0. Ativado/zerado por software para Ligar/Desligar o Timer 0.					
IE1	TCON.3	Flag de borda da interrupção Externa 1. Ativado por hardware quando uma borda na Interrupção Externa 1 é detectada. Zerado por hardware quando a interrupção é processada.					
IT1	TCON.2	Bit de controle da Interrupção Externa 1. Ativado/zerado por software para especificar se a Interrupção Externa 1 é sensível à descida de borda/nível baixo.					
IE0	TCON.1	Flag de borda da Interrupção Externa 0. Ativado por hardware quando uma borda na Interrupção Externa 0 é detectada. Zerado por hardware quando a interrupção é processada.					
IT0	TCON.0	Bit de controle da Interrupção Externa 0. Ativado/zerado por software para especificar se a Interrupção Externa 0 é sensível à descida de borda/nível baixo.					

SETB TR1 ; Dispara(liga) o T1

SETB TR0 ; Dispara(liga) o T0

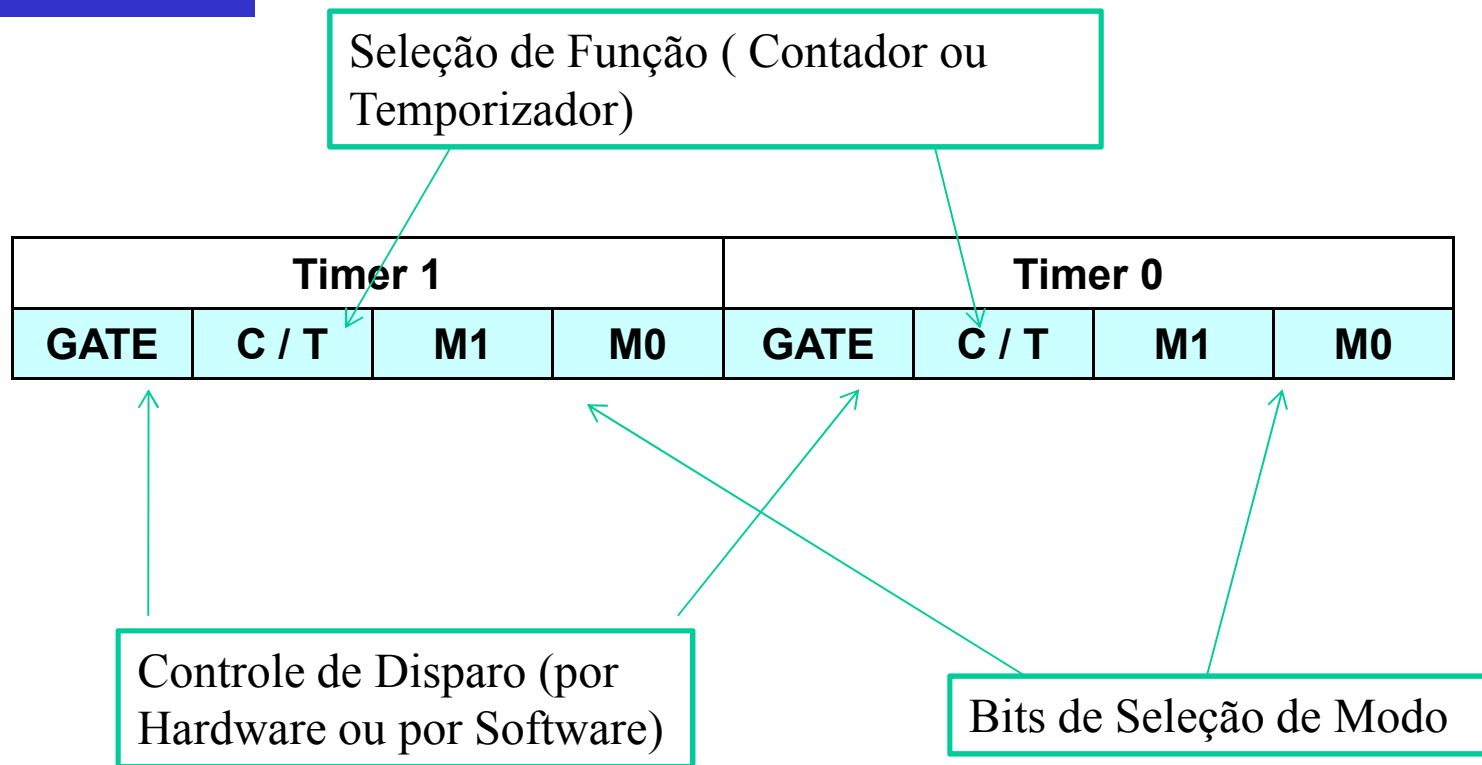
CLR TR1 ; Pára (Desliga) o T1

CLR TR0 ; Pára (Desliga) o T0

Temporizadores e Contadores (Timer/Counter)

Registrador TMOD

(Timer/Counter Mode) → **Não endereçável a Bit**



Controle de Disparo do Timer

a) Controle por Hardware

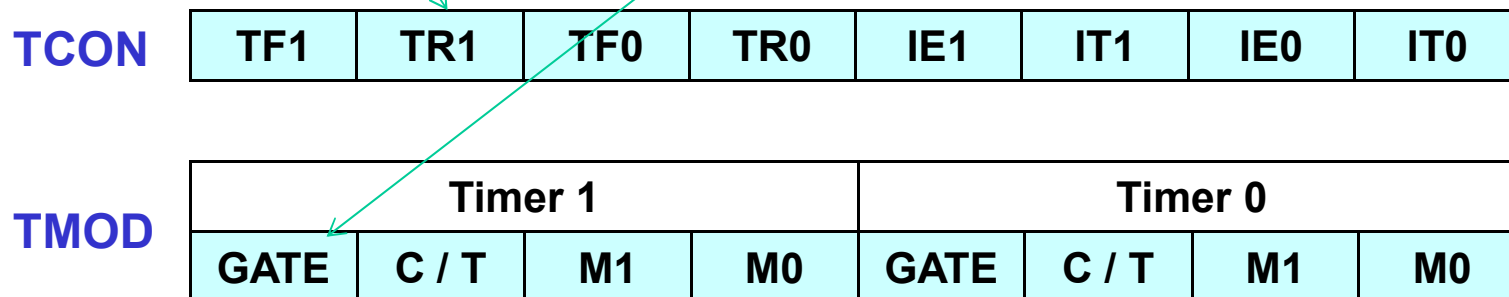


GATE = 1

Quem **liga e desliga** o Timer é um pulso aplicado externamente à CPU no pino Int1.

Exemplo para o Timer 1:

Quando o Bit TR1 (em TCON) é 1 e GATE=1 (em TMOD), o Timer1 contará enquanto o pino INT1 na CPU for 1.

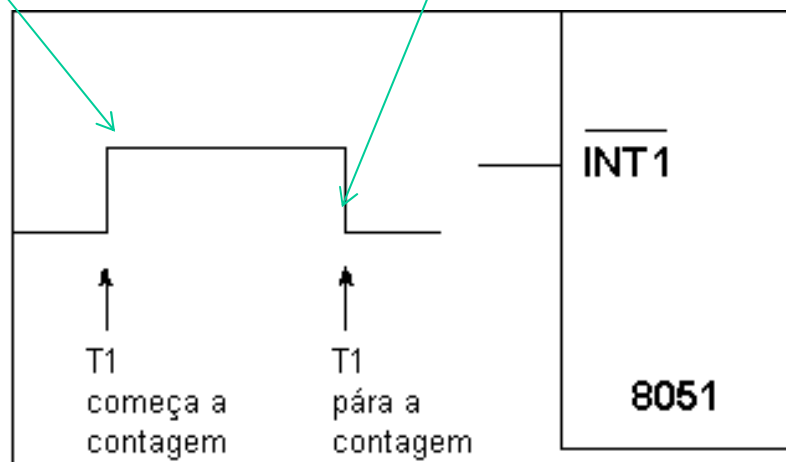


Exemplo para o Timer1:

GATE = 1

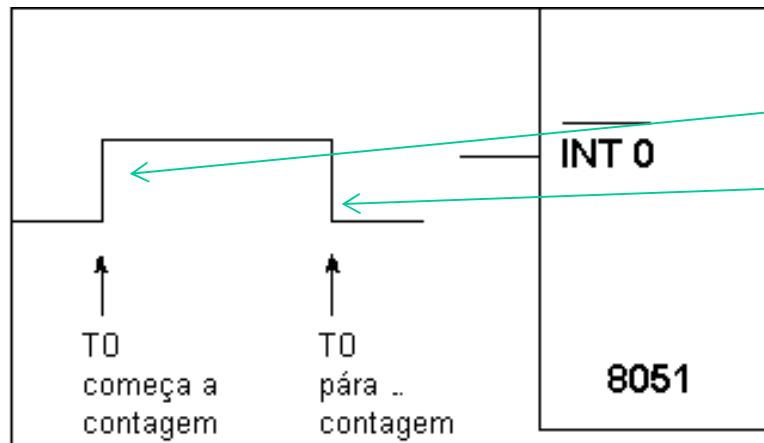
```
MOV TMOD,#10000000b ; Programa o Timer1 para
                       ; controle por Hardware (GATE=1)
SETB TR1              ; Liga o Timer1
```

Quando o valor lógico no pino Int1 da CPU for 1, o Timer1 começa a contar os pulsos do clock interno. Quando este valor for 0, o Timer1 pára a contagem.



Exemplo para o Timer0:

GATE = 1



Quando o valor lógico no pino Int0 da CPU for 1, o Timer1 começa a contar os pulsos do clock interno. Quando este valor for 0, o Timer0 pára a contagem.

	Timer 1				Timer 0			
TMOD	GATE	C / T	M1	M0	GATE	C / T	M1	M0

```
MOV TMOD,#00001000b ; Programa o Timer0 para  
; controle por Hardware (GATE=1)  
SETB TR0 ; Liga o Timer0
```


Controle de Disparo do Timer

a) Controle por Software



GATE = 0

Quem liga e desliga o Timer são os comandos aplicados sobre o bit TR (TR1 ou TR0).

	Timer 1				Timer 0			
TMOD	GATE	C / T	M1	M0	GATE	C / T	M1	M0

Exemplo para o Timer1 :

```
MOV  TMOD,#0xxxxxxx  ; programa o T1 para controle
                          ; por Software (GATE=0)

SETB TR1                ; Liga o T1

      |
      |

CLR  TR1                ; Desliga o T1
```

Exemplo para o Timer0:

GATE = 0

```
MOV  TMOD,#xxxx0xxxb ; programa o Timer0 para controle por
                               ;Software
SETB  TR0                ; Liga o Timer0
|
|
CLR  TR0                ; Desliga o Timer0
```

TMOD

Timer 1				Timer 0			
GATE	C / T	M1	M0	GATE	C / T	M1	M0

Seleção de Função

TMOD

Timer 1				Timer 0			
GATE	C / \bar{T}	M1	M0	GATE	C / \bar{T}	M1	M0

$C / \bar{T} = 0$ (Timer) – Temporizador

$C / \bar{T} = 1$ (Counter) - Contador

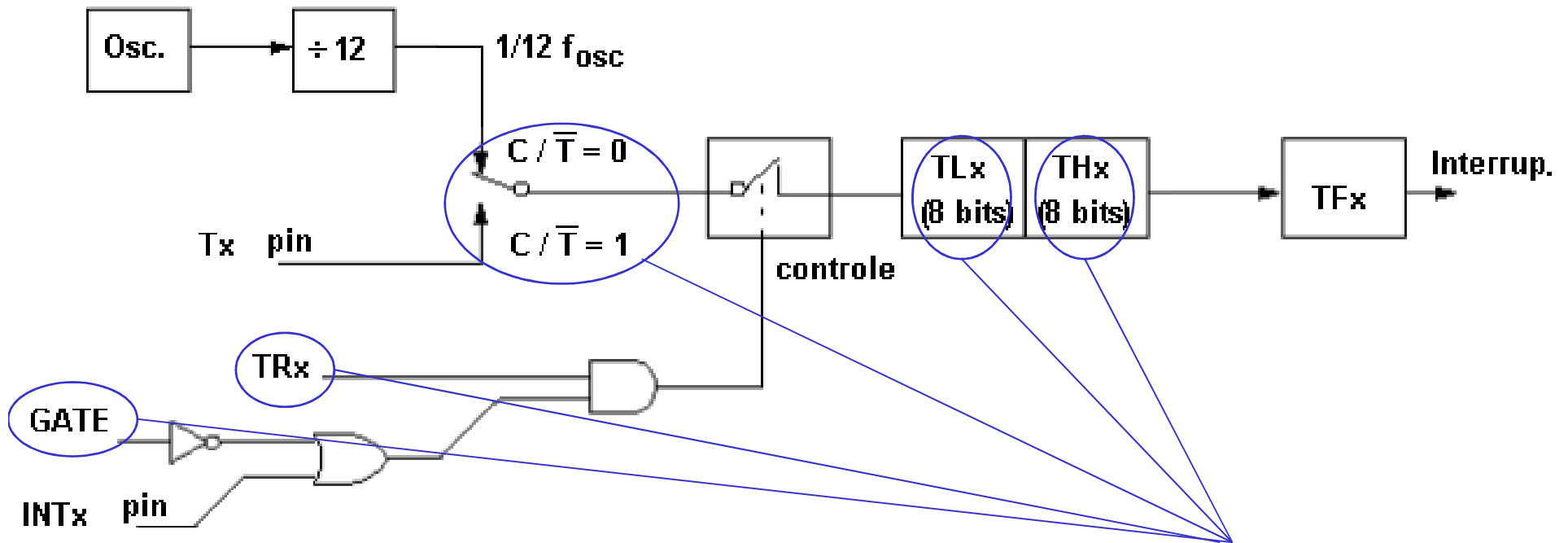
Timer (Temporizador) → entrada de pulsos através do clock interno ((frequência do oscilador)/12).

Counter (Contador) → entrada de pulsos externa através dos pinos T0 ou T1 da CPU.

Obs: A frequência externa deve ser no máximo a metade da frequência da CPU.

Temporizadores e Contadores (Timer/Counter)

Esquemático geral simplificado



**programáveis
por software**

Interrupção dos Timers/Counters

Obs: Os bits de flag de TCON, TF1 e TF0 são ativados e desativados por Hardware.

NÃO devem ser alterados por software!!!!

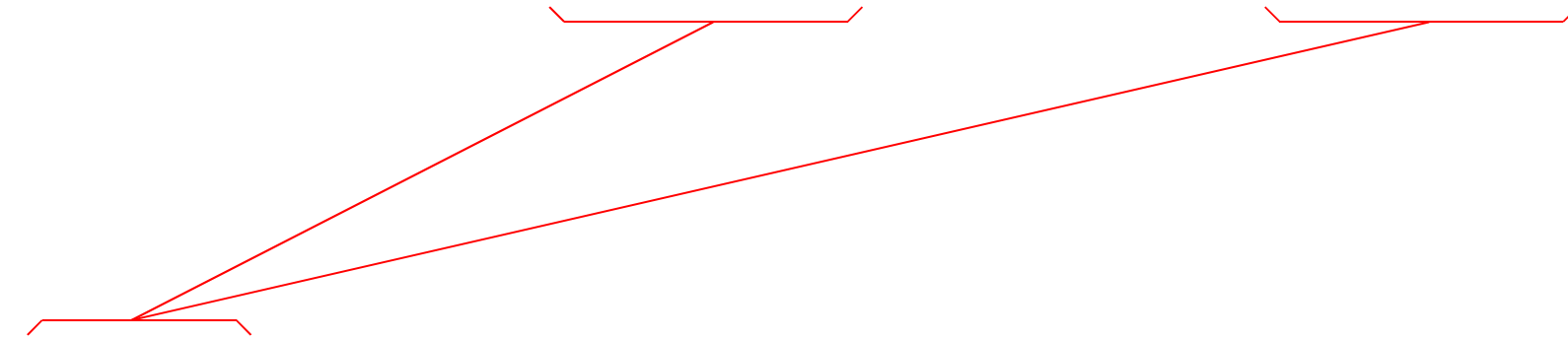
Primeiro Endereço	0033h
Interrupção Extra	002Bh
Interrupção da Serial	0023h
Overflow do Timer 1	001Bh
Interrupção Externa 1	0013h
Overflow do Timer 0	000Bh
Interrupção Externa 0	0003h
Reset	0000h

TF1	TCON.7	Flag de overflow do Timer 1. Ativado por hardware quando o Timer 1 transborda. Zerado por hardware assim que o processador salta para a rotina de atendimento da interrupção
TR1	TCON.6	Bit de controle do Timer 1. Ativado/zerado por software para Ligar/Desligar o Timer 1.
TF0	TCON.5	Flag de overflow do Timer 0. Ativado por hardware quando o Timer 0 transborda. Zerado por hardware assim que o processador salta para a rotina de atendimento da interrupção
TR0	TCON.4	Bit de controle do Timer 0. Ativado/zerado por software para Ligar/Desligar o Timer 0.

Seleção de Modo

TMOD

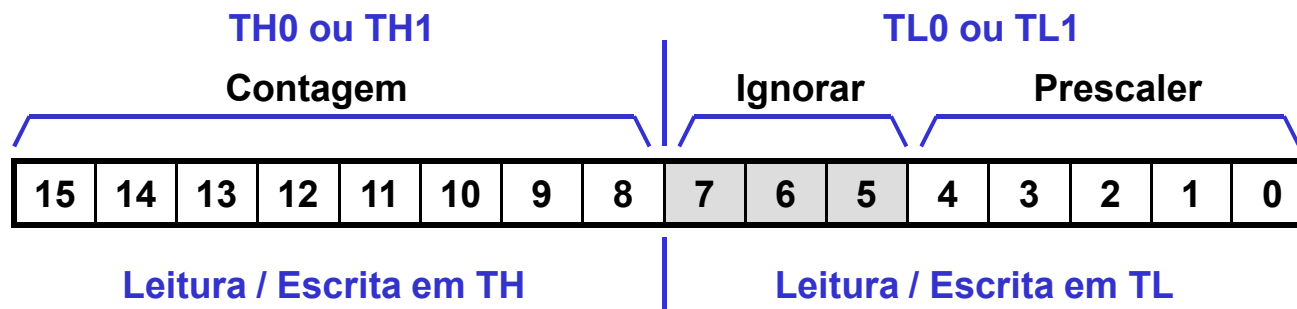
Timer 1				Timer 0			
GATE	C / \bar{T}	M1	M0	GATE	C / \bar{T}	M1	M0



M1	M0	MODOS DE OPERAÇÃO	
0	0	0	Contador / Temporizador de 13 bits (8 bits + 5 bits de prescaler)
0	1	1	Contador / Temporizador de 16 bits
1	0	2	Contador / Temporizador de 8 bits auto-recarregável
1	1	3	Para o T0: TL0 é um Contador / Temporizador de 8 bits controlado pelos bits de controle do T0 e TH0 é um Contador / Temporizador de 8 bits controlado pelos bits de controle do T1
1	1	3	O T1 não funciona no modo 3

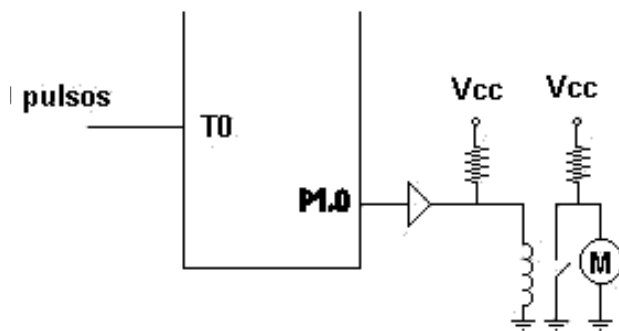
Modo 0

Contador / Temporizador de 13 Bits → 8 Bits + 5 Bits de prescaler



Exemplo :

Usar o T0 como contador no Modo 0, para contar 32 pulsos 100 vezes e, após a contagem acionar um mecanismo que liga um motor conectado ao bit 0 da porta P1.



Para contar 100 pulsos em TH0, o contador deve contar de x até 255.

$$255 - x = 100$$



$$x = 155 = 9Bh$$

O Contador é crescente (UP Counter) → O contador contará em TH0 de 155 a 255, ou seja, 100 vezes.

100 contagens → 9Bh até FFh

(FFh – 9Bh = 64h → 100d)

Logo, deve-se carregar inicialmente o valor de TH0 com 9Bh

```
MOV TH0,#9Bh
```


O T0 conta em TL0 32 pulsos . A cada 32 pulsos é incrementado o valor de TH0.

Depois de 100 x 32 pulsos, ocorre um overflow em TH0 ocasionando TF0=1 .

TCON

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

TMOD

Timer 1				Timer 0			
GATE	C / \bar{T}	M1	M0	GATE	C / \bar{T}	M1	M0
0	0	0	0	0	1	0	0

Exemplo:

```
ORG 0
SJMP Prog
;*****
Sub1:  ORG 000Bh      ; Endereço da sub-rotina do T0
      CLR EA        ; Evita Interrupção da Interrupção
      SETB P1.0     ; Aciona o mecanismo
      CLR P1.0
      SETB EA       ; Re-habilita as Interrupções
      MOV TH0,#9Bh  ; contar cem vezes
      RETI          ;Retorna
;*****
Prog:  SETB ET0      ;habilita Interrupção do T0
      MOV TH0,#9Bh  ; contar cem vezes
      MOV TL0,#00h  ; contar trinta e dois pulsos
      MOV TMOD,#04  ;T0 no modo 0 como Contador e
                   ;disparo por software
      SETB EA       ;habilita as interrupções
      SETB TR0      ;Dispara o T0
      CLR P1.0      ; Desliga o mecanismo
Loop:  SJMP Loop     ; Permanece em Loop
      END
```

Obs: Toda a vez que a contagem atingir 100 x 32 pulsos o programa desvia para Sub1 e aciona o mecanismo

Modo 1

Contador/Temporizador de 16 Bits

- O valor inicial pode ser programado por Software

Exemplo 1:

- 1) Programar o T1 como um **Temporizador** (Timer) no Modo 1 com controle por Software para gerar um pulso positivo de 30 ms na Porta P1.0.
(Se o Cristal for de 12 MHz, a frequência será $12 \text{ MHz}/12 = 1 \text{ MHz}$)

Programação do T1

período = 1 us = 1 ciclo de Máquina

TMOD	Timer 1				Timer 0			
	GATE	C / \bar{T}	M1	M0	GATE	C / \bar{T}	M1	M0
	0	0	0	1	0	0	0	0

Frequência do temporizador (ft)

$$ft = \frac{f}{12}$$

Frequência do Cristal (f)

Intervalo de tempo de contagem (Δt)

$$\Delta t = n \cdot \frac{1}{ft}$$

Número de pulsos a serem contados (n)

O contador deve contar de x até o máximo valor 65535 (FFFFh)

$$65535 - x = n$$

Logo,

$$65535 - x = \Delta t \cdot ft = \Delta t \cdot \frac{f}{12} \Rightarrow x = 65535 - \Delta t \cdot \frac{f}{12}$$

O valor de x deve ser arredondado para o inteiro mais próximo

$$x = 65535 - \Delta t \cdot \frac{f}{12}$$

$\Delta t \rightarrow \mu\text{s}$

$f \rightarrow \text{MHz}$

$$\Delta t = 30 \text{ ms} = 30.000 \mu\text{s}$$

$$f = 12 \text{ MHz}$$

$$x = 65535 - 30000 \cdot \frac{12}{12}$$

$$x = 35535 = 8ACFh$$

30 ms \rightarrow contar de 8ACFh a FFFFh (35535 a 65535 = 30.000 us) se o cristal for de 12 MHz

```
MOV TH1,#8Ah
MOV TL1,#0CFh
```

Pulso positivo de 30ms na Porta P1.0

```
ORG 0
SJMP Prog
; *****
Sub1: ORG 001Bh ; Sub-rotina do T1
      CLR EA
      CLR P1.0 ; Fim do Pulso depois de 30 ms
      RETI
; *****
Prog: CLR P1.0 ; Zera a saída de pulso
      SETB ET1 ; Habilita a interrupção do T1
      MOV TMOD,#10h ; T1 no Modo 1, temporizador e
                   ; controle por software
      MOV TH1,#8Ah ; Contar de 8ACFh até FFFFh
      MOV TL1,#0CFh
      SETB EA ; habilita as interrupções
      SETB TR1 ; Inicia a contagem
      SETB P1.0 ; Início do pulso
      SJMP $
END
```

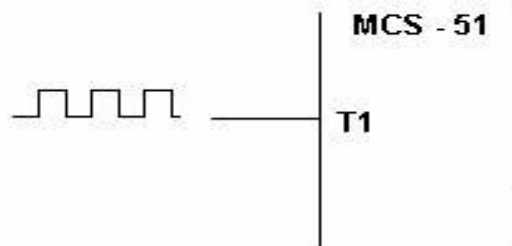
Exemplo 2:

Programar o T1 como Contador (Counter) no Modo 1 com controle por Software. A entrada de pulsos deverá ser aplicada em T1.

Programação do T1

TMOD	Timer 1				Timer 0			
	GATE	C / \bar{T}	M1	M0	GATE	C / \bar{T}	M1	M0
	0	1	0	1	0	0	0	0

Os pulsos a serem contados devem entrar através do pino T1 da CPU em frequência inferior à metade do Cristal.

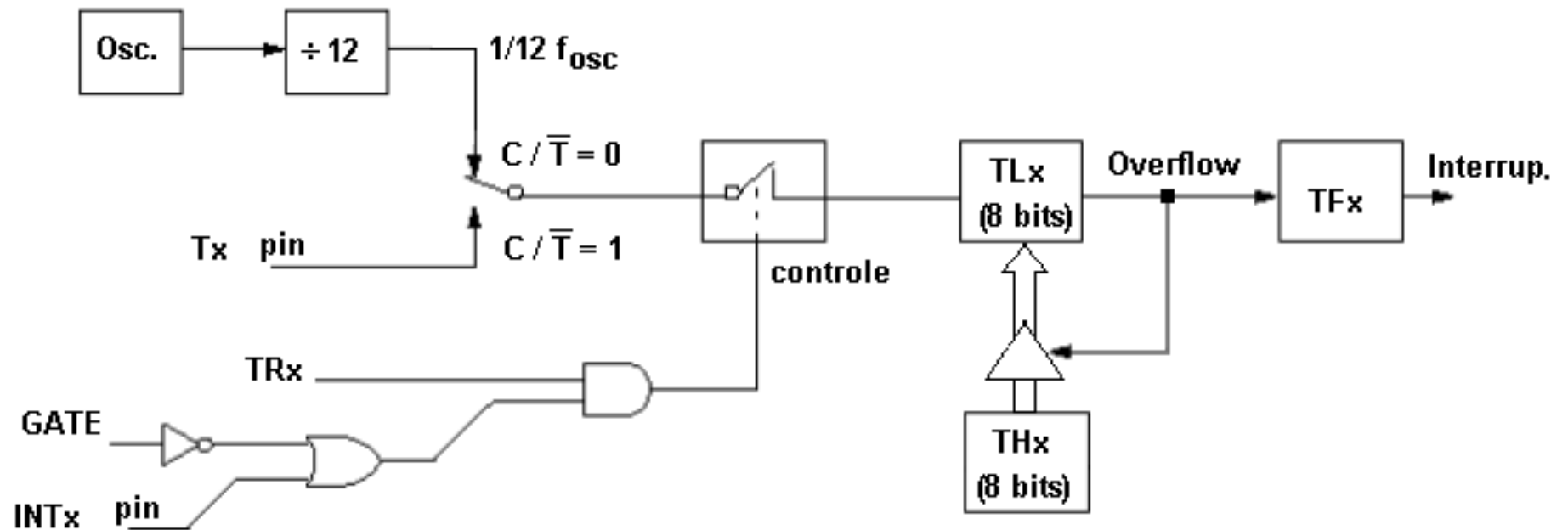


```
MOV    TMOD,#50h    ;  
SETB   TR1          ; Inicia a contagem
```

O resultado da contagem pode ser lido em TH1(MSB) e TL1(LSB)

Modo 2

Contador/Temporizador de 8 Bits Auto-recarregável



Deve-se armazenar em THx o Valor de Recarga.

Exemplo:

1) Temporizador de 8 Bits no Modo 2 (de 7Fh a FFh) para o T1. Controle por software.

Programação do T1

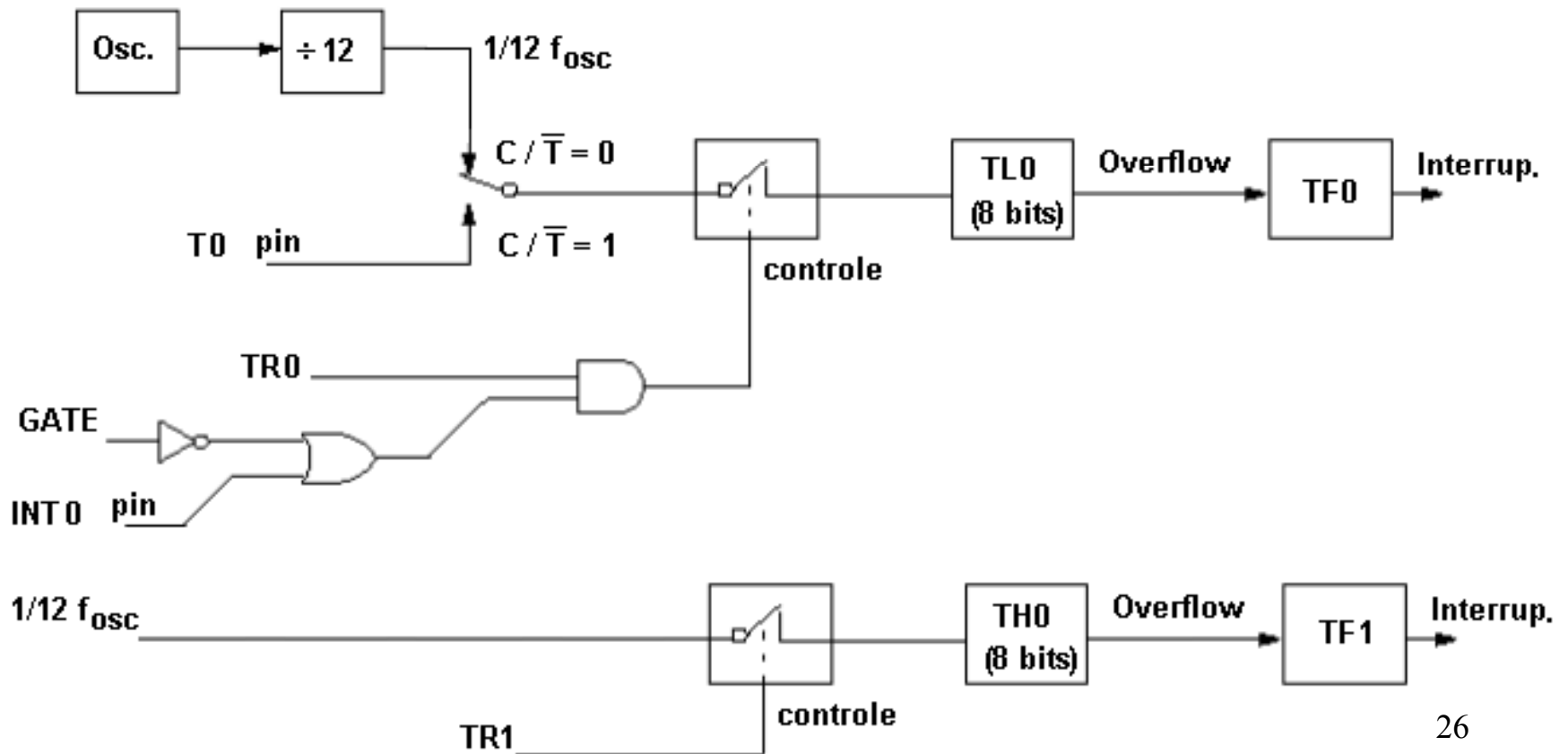
TMOD	Timer 1				Timer 0			
	GATE	C / \bar{T}	M1	M0	GATE	C / \bar{T}	M1	M0
	0	0	1	0	0	0	0	0

```
MOV    TMOD,#20h           ; Programa o T/C 1 no Modo 2 como  
                                     ; Temporizador e controle por Software  
  
MOV    TH1,#7Fh  
  
MOV    TL1,#7Fh  
  
SETB   TR1                   ; Inicia a contagem
```

Quando o T1 chegar em FFh, recomeça de 7Fh (valor de TH1).

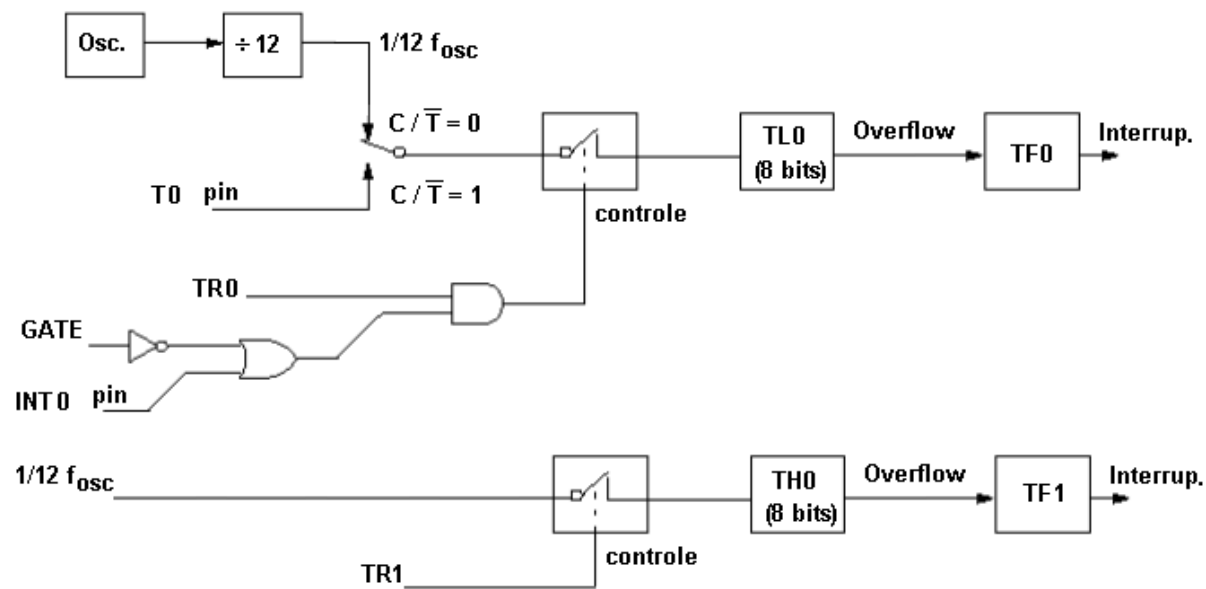
Modo 3

- O T1 não funciona no Modo 3.
- O T0 no Modo 3 estabelece TL0 e TH0 como dois contadores separados de 8 Bits cada.



Modo 3

- Como o T1 não opera neste modo, TH0 gera uma solicitação de Interrupção através de TF1.
- O T1 pode ser programado em outro modo, mas aí, não gera overflow em TF1 para solicitação de Interrupção .
- TH0 só funciona como Temporizador (Timer) pois só usa a frequência interna (1/12 oscilador)



Exemplo: Programa que gera duas ondas quadradas usando as interrupções dos Timers

Os Contadores contam Ciclos de Máquina !!!!

```

ORG 0
SJMP Prog
; *****
ORG 000BH
SJMP Sub0
; *****
ORG 001Bh
SJMP Sub1
; *****
Prog: CLR P1.0
      CLR P1.1
      SETB ET1 ←
      SETB ET0
      MOV TMOD, #11h
      MOV TH1, #0FFh
      MOV TL1, #0FEh
      MOV TH0, #0FFh
      MOV TL0, #0FDh
      SETB EA
      SETB TR1
      SETB TR0
      SJMP $
; *****

```

```

Sub0: CLR EA
      CPL P1.0
      CLR TR1 ←
      CLR TR0
      MOV TH0, #0FFh
      MOV TL0, #0FDh
      SETB EA
      SETB TR1
      SETB TR0
      RETI
; *****
Sub1: CLR EA
      CPL P1.1 ←
      CLR TR1 ←
      CLR TR0
      MOV TH1, #0FFh
      MOV TL1, #0FEh
      SETB EA
      SETB TR1
      SETB TR0
      RETI
; *****
END

```

Exercício Aula_5

- Escrever o programa do exemplo anterior no simulador MCU8051.
- Inserir Breakpoints nas linhas (**clique no número da linha**) 12 , 27 e 39 (instruções logo após os bits das portas terem sido alterados. Anotados com setas na figura anterior).
- Avaliar a forma de onda nas duas portas (desenhar) e marcar o valor do tempo (em microssegundos) das partes positiva e negativa de cada uma.
- Considerar o cristal de 12 MHz.