

# Arquitetura de Computadores

---

PROF. ISAAC

# Contador e Temporizador

---

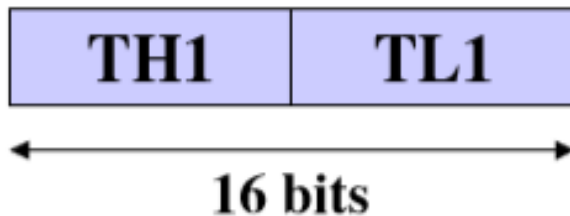
# Contador/temporizador – CT0 e CT1

A família 8051 disponibiliza dois contadores, denominados contador/temporizador 0 e contador/temporizador 1 (CT0 e CT1).

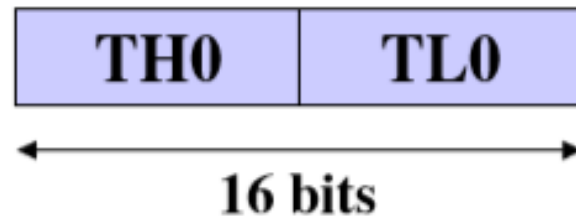
Eles são construídos com dois contadores binários independentes, de 16 bits.

Abaixo os 4 registradores:

**Contador/Temporizador 1**



**Contador/Temporizador 0**



# Contador/temporizador – CT0 e CT1

Esses bytes TH0, TL0, TH1 e TL1, podem ser lidos ou escritos a qualquer momento. Tais contadores binários contam de forma ascendente e, a cada vez que ultrapassam seu limite, que é o valor 65.535, a flag de overflow é ativada.

O interessante é que essa flag, além de poder ser consultada a qualquer instante, também pode provocar interrupção.

**Contador/Temporizador 1**



← 16 bits →

**Contador/Temporizador 0**

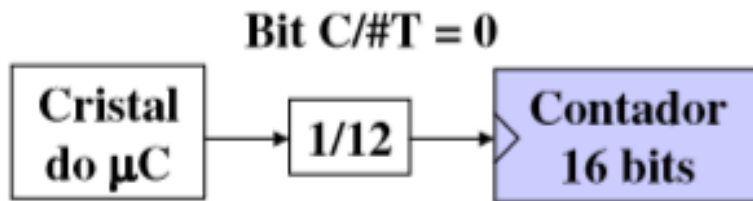


← 16 bits →

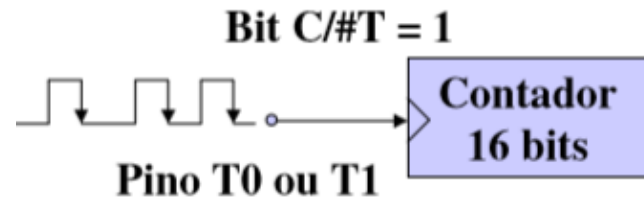
# Contador/temporizador – CT0 e CT1

CT0 e CT1 podem operar como contador ou temporizador.

- Para operar como temporizador ele utiliza o clock do Cristal.
- Para operar como contador ele utiliza os pinos T0 e T1.

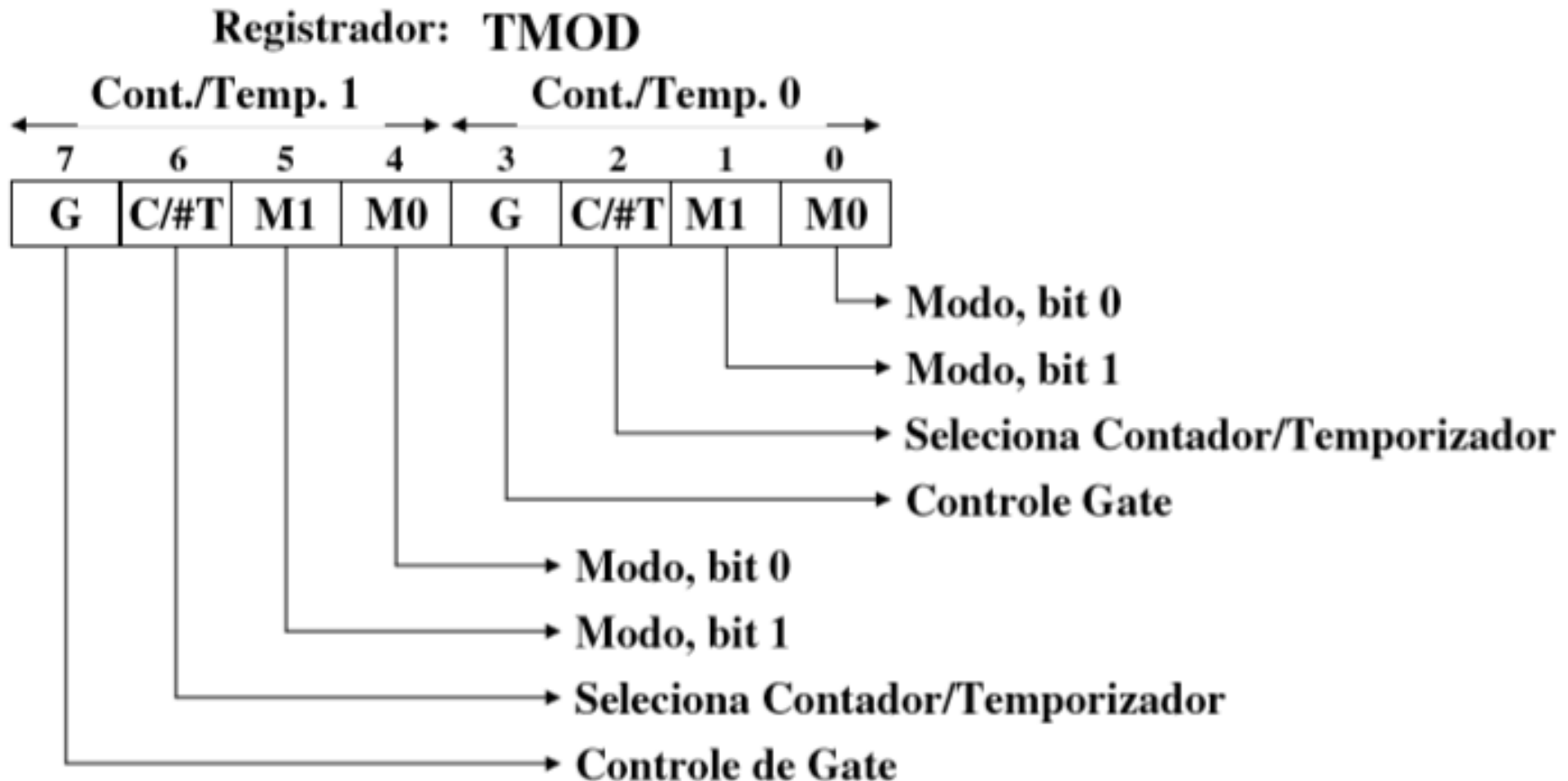


*Operando como TEMPORIZADOR*



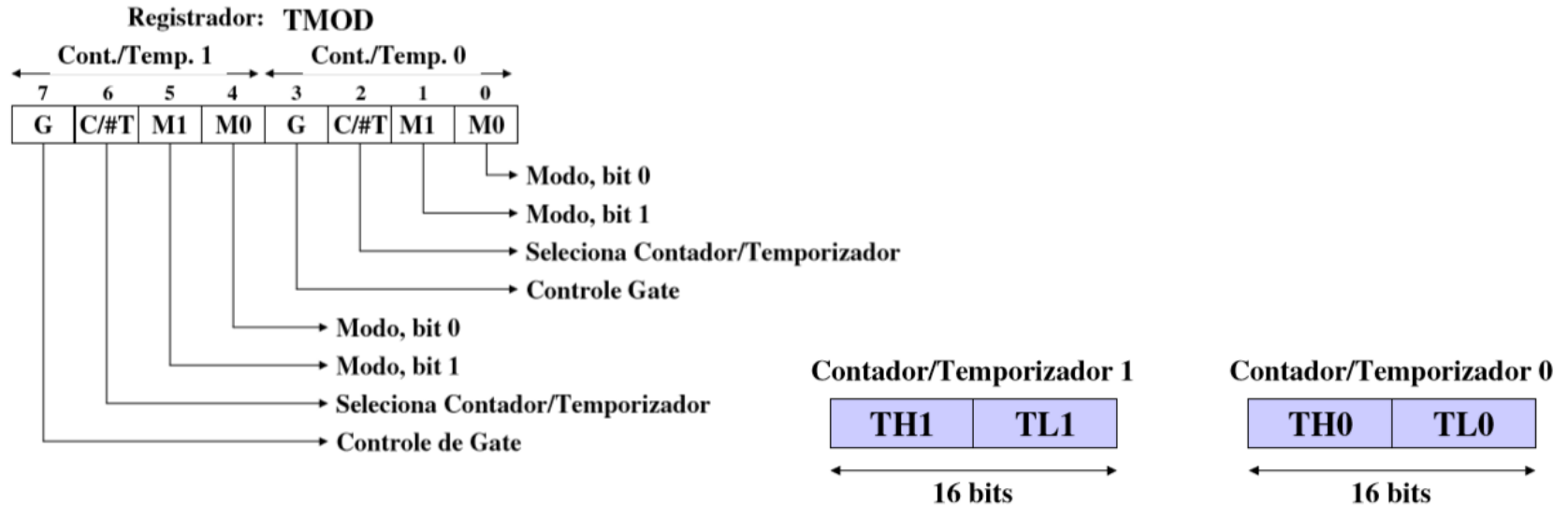
*Operando como CONTADOR*

# Registrador: TMOD



Descrição dos bits do registrador TMOD.

# Registrador: TMOD



M1	M0	Modo	Descrição
0	0	0	THi é CT de 8 bits e TLi é pré-escala de 5 bits.
0	1	1	THi e TLi formam CT de 16 bits.
1	0	2	TLi é CT de 8 bits e THi armazena valor de recarga.
1	1	3	TL0 é CT de 8 bits (usando TR0, #INT0 e TF0), TH0 é CT de 8 bits (usando TR1, #INT1 e TF1) TH1 e TL1 parado (pode operar em outros modos).

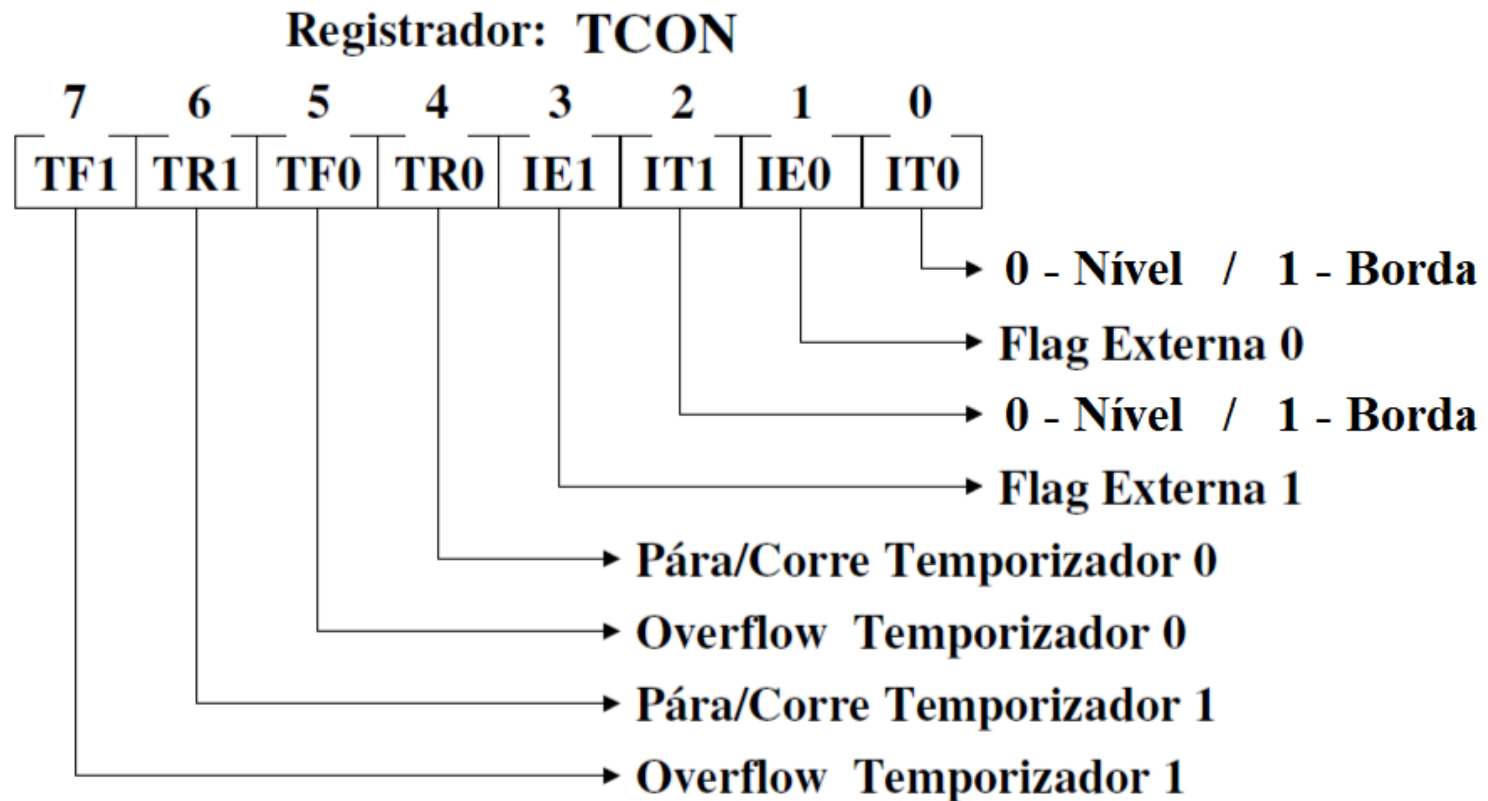
# Registrador: TMOD

Bit	T/C	Nome	Descrição
7	T/C1	GATE1	Modo de controle (0: software, 1: hardware)
6		C/T1	Seleciona timer (0) ou counter (0)
5		T1M1	Seleção de modo do timer 1
4		T1M0	
3	T/C0	GATE0	Modo de controle (0: software, 1: hardware)
2		C/T0	Seleciona timer ou counter
1		T0M1	Seleção de modo do timer 0
0		T0M0	

O bit GATE quando ativado permite que um sinal externo aplicado ao pino #INTi ligue ou desligue o contador.



# Registrador: TCON (Timer Controller)



Registrador TCON, onde se especifica se as interrupções externas trabalharão por nível ou por flanco.

# Registrador: TCON (Timer Controller)

Os quatro *bits* menos significativos do registrador de controle dos *timers*/contadores chamado de TCON gerenciam o funcionamento das interrupções externas.

<i>bits</i>	7	6	5	4	3	2	1	0
	TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

São 2 entradas de interrupção externas:

- interrupção 0 (P3.2/ INT0)
- interrupção 1 (P3.3/ INT1)

# Registrador: TCON (Timer Controller)

<i>bits</i>	7	6	5	4	3	2	1	0
	TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

## **TR0** (TCON.4) e **TR1** (TCON.6)

É o *bit* de controle responsável por fazer o *timer*/contador contar ou parar a contagem (**run/stop**).

## **TF0** (TCON.5) e **TF1** (TCON.7)

É o *bit* que indica o estouro do *timer*/contador (*overflow*).

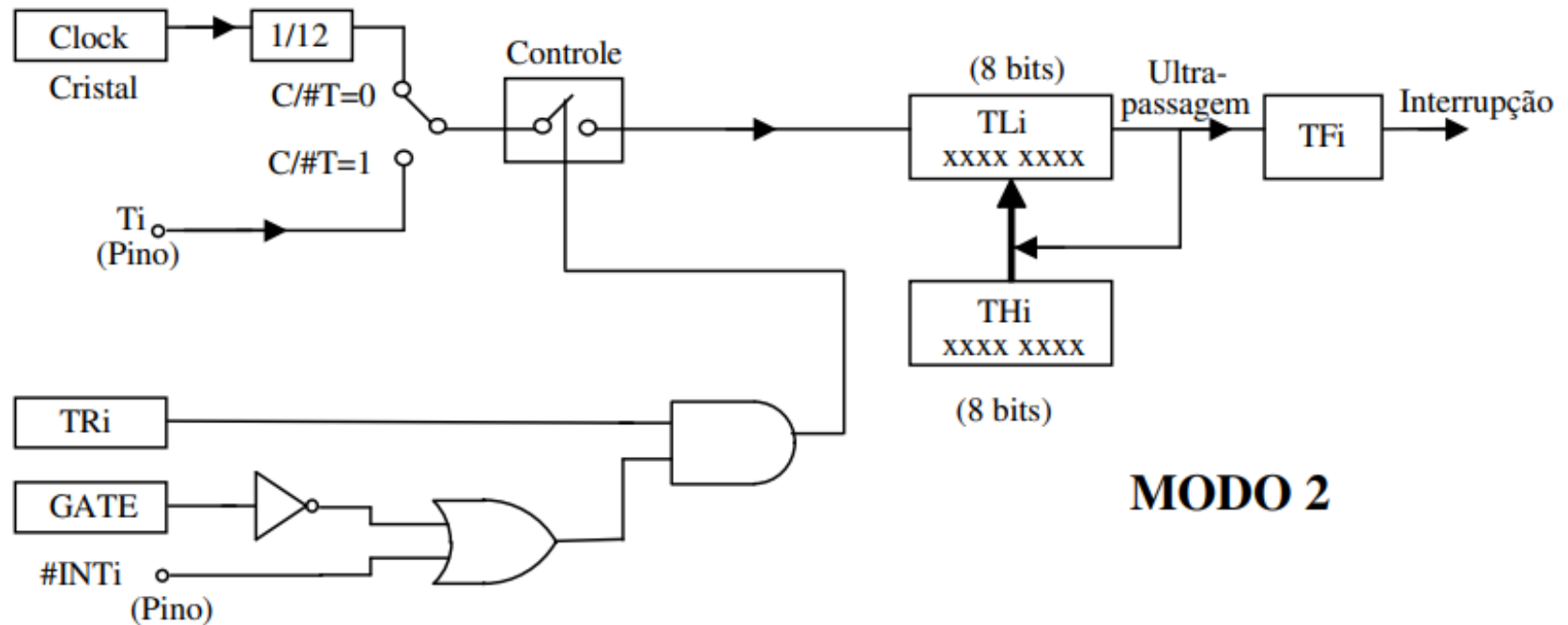
Ele é setado por *hardware* sempre que o *timer*/contador estoura (*overflow*).

Esse *flag* é resetado por *hardware* sempre que a sub-rotina de atendimento a essa fonte de interrupção é vetorizada e executada.

# Exemplo com temporizador no Modo 2

---

# Modo 2



# Exemplo 01

---

# Exemplo 1

---

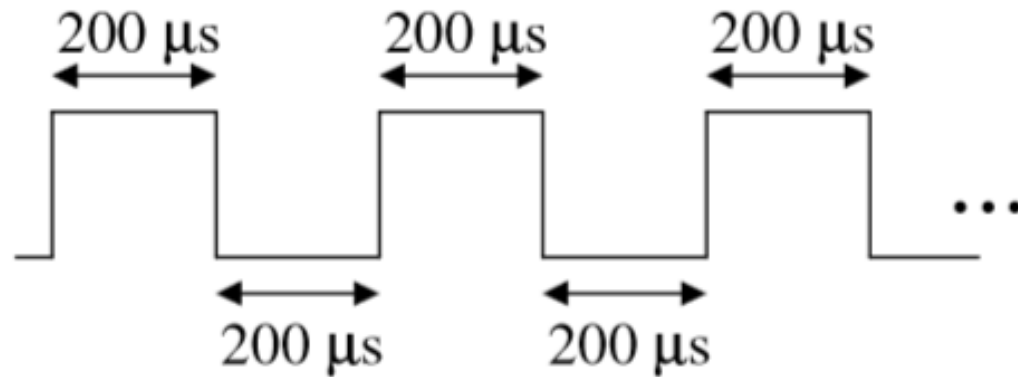
Usando o Temporizador, gere pelo pino P1.0 uma onda quadrada de 2,5 kHz

# Exemplo 1

## Solução:

---

O período de uma onda quadrada de 2,5 kHz é de  $400\ \mu\text{s}$ , o que significa que o pino P1.0 deve, repetitivamente, ficar  $200\ \mu\text{s}$  em nível baixo e  $200\ \mu\text{s}$  em nível alto.





# Exemplo 1

## Solução:

---

Vamos programar o CT0 para ativar o bit de ultrapassagem (*overflow*) TF0 a cada 200  $\mu$ s. Cada vez que TF0 for igual a 1, invertamos o pino P1.0 e apagamos TF0 para que se possa ativá-lo novamente no próximo *overflow*.

Usaremos o modo 2, onde o contador deve ser preparado para sofrer a ultrapassagem após 200 contagens e, como a contagem é crescente, ele deve ser programado com o valor  $256 - 200 = 56$ .

A contagem sempre começa em 56 e a ultrapassagem acontece após 200 contagens ( $255 + 1$  que ocorre overflow).

# Exemplo 1

## Solução:

---

### CONFIG:

<b>MOV TMOD,#2</b>	<b>;CT0 no modo 2</b>
<b>MOV TH0,#56</b>	<b>;valor para a recarga</b>
<b>MOV TL0,#56</b>	<b>;valor para a primeira contagem</b>
<b>SETB TR0</b>	<b>;liga o CT0</b>

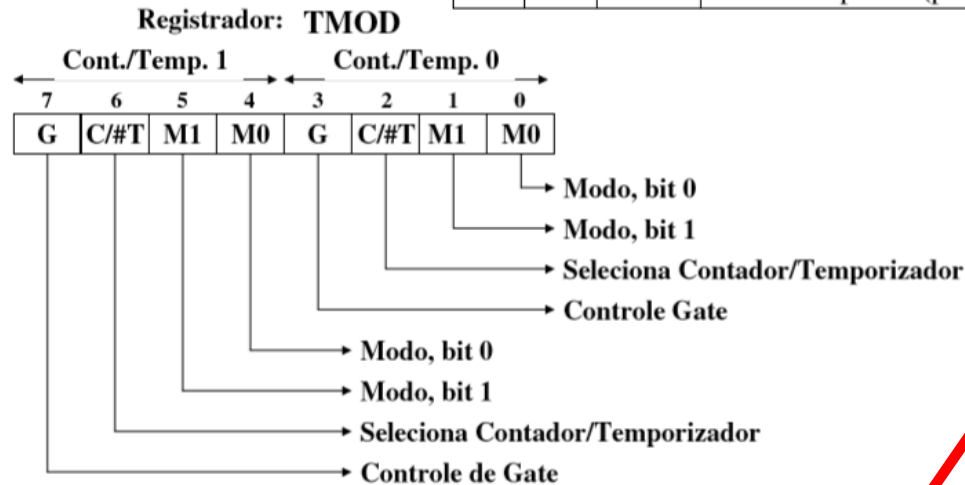
### ROT:

<b>JNB TF0,\$</b>	<b>;aguarda bit de ultrapassagem</b>
<b>CLR TF0</b>	<b>;apaga bit de ultrapassagem</b>
<b>CPL P1.0</b>	<b>;complementa P1.0</b>
<b>SJMP ROT</b>	<b>;fecha o laço</b>

# Exemplo 1

## Solução:

M1	M0	Modo	Descrição
0	0	0	THi é CT de 8 bits e TLi é pré-escala de 5 bits.
0	1	1	THi e TLi formam CT de 16 bits
1	0	2	TLi é CT de 8 bits e THi armazena valor de recarga.
1	1	3	TL0 e CT de 8 bits (usando TR0, #INT0 e TF0), TH0 é CT de 8 bits (usando TR1, #INT1 e TF1) TH1 e TL1 parado (pode operar em outros modos)



## CONFIG:

```
MOV TMOD,#2           ;CT0 no modo 2
MOV TH0,#56           ;valor para a recarga
MOV TL0,#56           ;valor para a primeira contagem
SETB TR0              ;liga o CT0
```

## ROT:

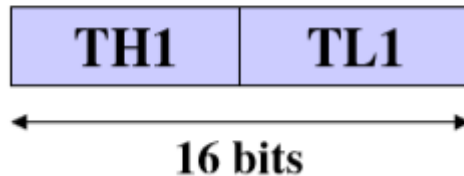
```
JNB TF0,$             ;aguarda bit de ultrapassagem
CLR TF0               ;apaga bit de ultrapassagem
CPL P1.0              ;complementa P1.0
SJMP ROT              ;fecha o laço
```

# Exemplo 1

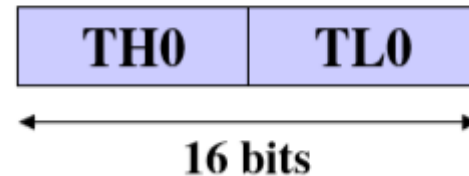
## Solução:

M1	M0	Modo	Descrição
0	0	0	THi é CT de 8 bits e TLi é pré-escala de 5 bits.
0	1	1	THi e TLi formam CT de 16 bits
1	0	2	TLi é CT de 8 bits e THi armazena valor de recarga.
1	1	3	TL0 e CT de 8 bits (usando TR0, #INT0 e TF0), TH0 é CT de 8 bits (usando TR1, #INT1 e TF1) TH1 e TL1 parado (pode operar em outros modos).

Contador/Temporizador 1



Contador/Temporizador 0



### CONFIG:

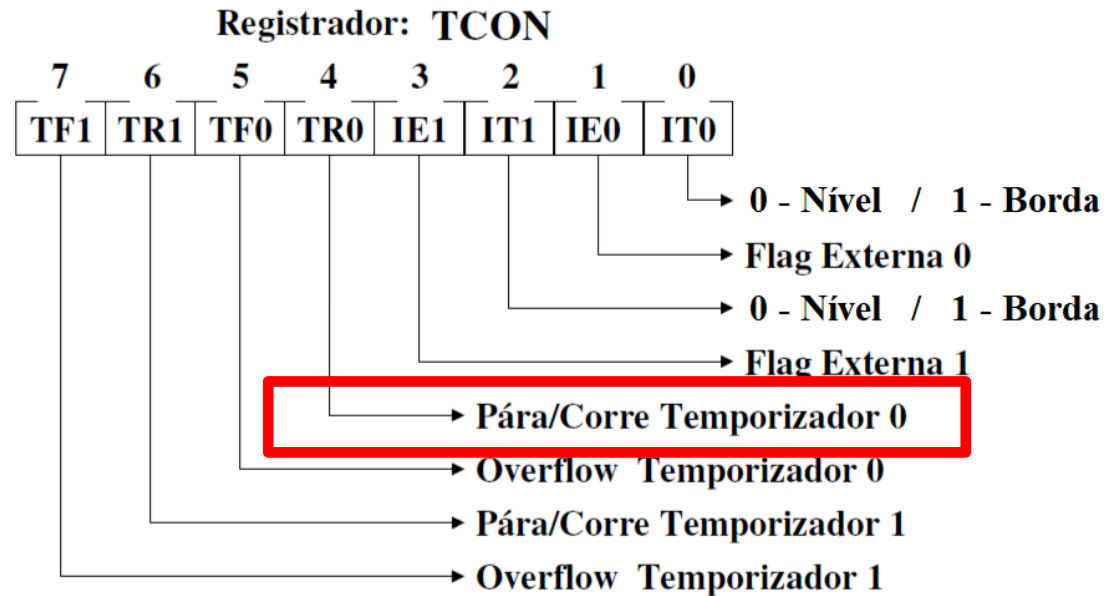
MOV TMOD,#2	:CT0 no modo 2
MOV TH0,#56	;valor para a recarga
MOV TL0,#56	;valor para a primeira contagem
SETB TR0	;liga o CT0

### ROT:

JNB TF0,\$	;aguarda bit de ultrapassagem
CLR TF0	;apaga bit de ultrapassagem
CPL P1.0	;complementa P1.0
SJMP ROT	;fecha o laço

# Exemplo 1

**Solução:**



**CONFIG:**

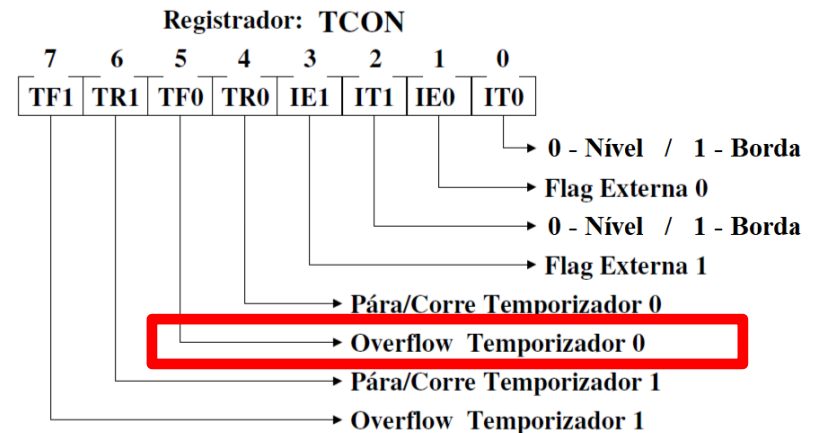
MOV TMOD,#2	;CT0 no modo 2
MOV TH0,#56	;valor para a recarga
MOV TL0,#56	;valor para a primeira contagem
SETB TR0	;liga o CT0

**ROT:**

JNB TF0,\$	;aguarda bit de ultrapassagem
CLR TF0	;apaga bit de ultrapassagem
CPL P1.0	;complementa P1.0
SJMP ROT	;fecha o laço

# Exemplo 1

## Solução:



Fica preso nesta linha enquanto TF0 for igual a zero.

### CONFIG:

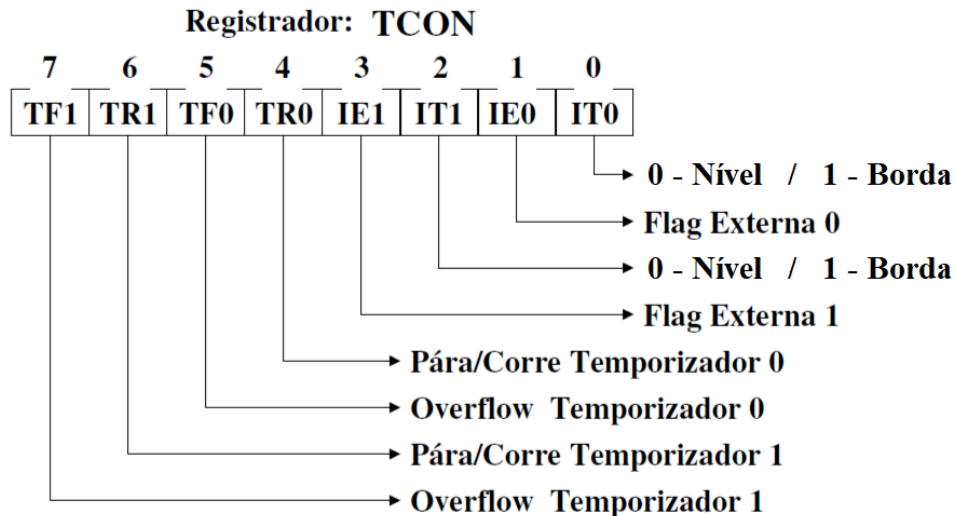
MOV TMOD,#2	;CT0 no modo 2
MOV TH0,#56	;valor para a recarga
MOV TL0,#56	;valor para a primeira contagem
SETB TR0	;liga o CT0

### ROT:

JNB TF0,\$	;aguarda bit de ultrapassagem ( <i>overflow</i> )
CLR TF0	;apaga bit de ultrapassagem
CPL P1.0	;complementa P1.0
SJMP ROT	;fecha o laço

# Exemplo 1

## Solução:



### CONFIG:

```
MOV TMOD,#2           ;CT0 no modo 2
MOV TH0,#56           ;valor para a recarga
MOV TL0,#56           ;valor para a primeira contagem
SETB TR0              ;liga o CT0
```

### ROT:

```
INR TF0,$             ;aguarda hit de ultrapassagem
CLR TF0               ;apaga bit de ultrapassagem (overflow)
CPL P1.0              ;complementa P1.0
SJMP ROT              ;pula incondicionalmente para ROT
```

# Exemplo 1

## Solução:

---

### CONFIG:

**MOV TMOD,#2**

**;CT0 no modo 2**

**MOV TH0,#56**

**;valor para a recarga**

**MOV TL0,#56**

**;valor para a primeira contagem**

**SETB TR0**

**;liga o CT0**

### ROT:

**JNB TF0,\$**

**;aguarda bit de ultrapassagem**

**CLR TF0**

**;apaga bit de ultrapassagem**

**CPL P1.0**

**;complementa P1.0**

**SJMP ROT**

**;pula incondicionalmente para ROT**

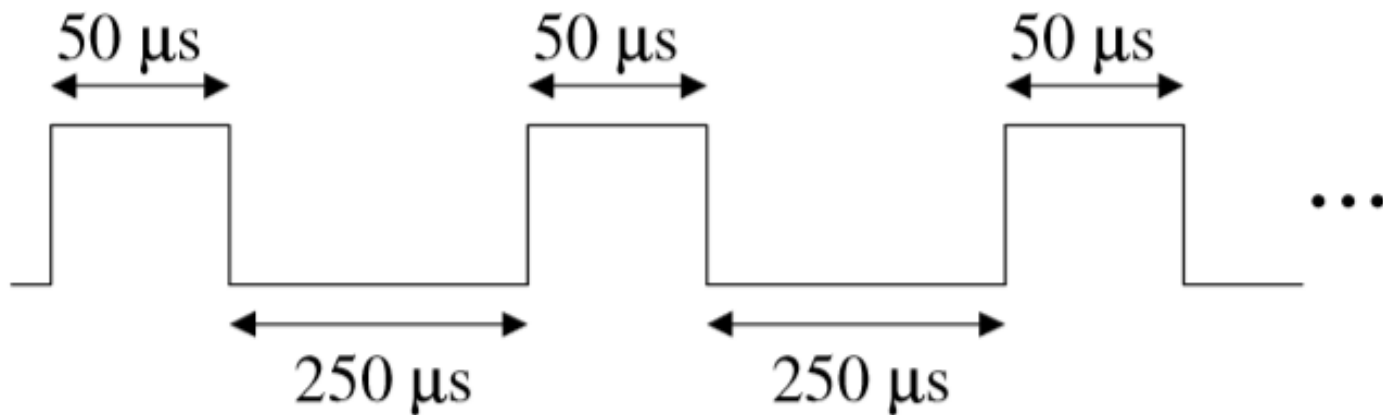


# Exemplo 02

---

## Exemplo 2

Usando o Temporizador, gere pelo pino P1.0 a forma de onda apresentada:



## Exemplo 2

### Solução:

---

Usaremos o modo 2, onde o contador deve ser preparado para sofrer a ultrapassagem após 50 contagens, portanto ele deve ser programado com o valor  $256 - 50 = 206$  e setar P1.0

Depois contador deve ser preparado para sofrer a ultrapassagem após 250 contagens, então ele deve ser programado com o valor  $256 - 250 = 6$  para dar clear P1.0

# Exemplo 2

## Solução:

### CONFIG:

<b>MOV TMOD,#2</b>	<b>;CT0 no modo 2</b>
<b>MOV TH0,#206</b>	<b>;valor para a recarga</b>
<b>MOV TL0,#206</b>	<b>;valor para a primeira contagem</b>
<b>SETB TR0</b>	<b>;liga o CT0</b>
<b>SETB P1.0</b>	<b>;inicia o primeiro período de 50 <math>\mu</math>s</b>

### LB1:

<b>JNB TF0,\$</b>	<b>;aguarda bit de ultrapassagem</b>
<b>MOV TL0,#6</b>	<b>;valor para a primeira contagem</b>
<b>CLR TF0</b>	<b>;apaga bit de ultrapassagem</b>
<b>CLR P1.0</b>	<b>;faz P1.0 = 0</b>

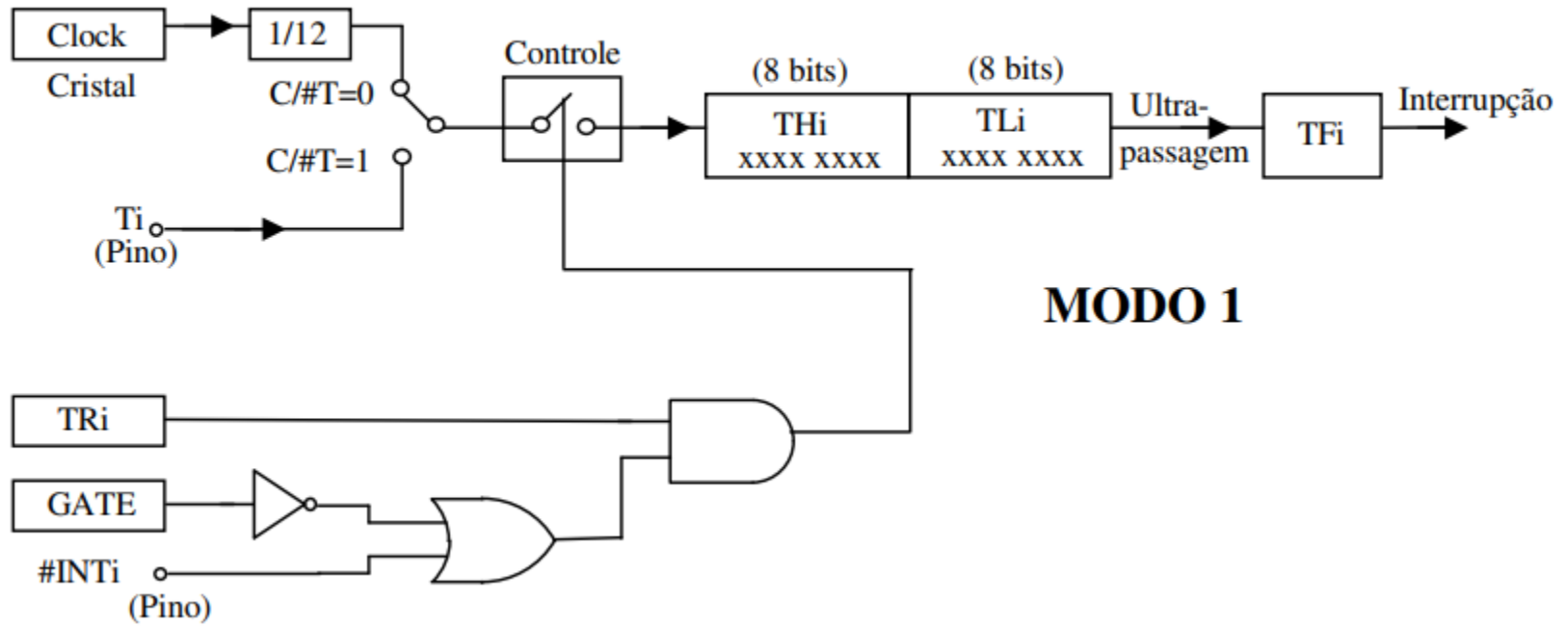
### LB2:

<b>JNB TF0,\$</b>	<b>;aguarda bit de ultrapassagem</b>
<b>CLR TF0</b>	<b>;apaga bit de ultrapassagem</b>
<b>SETB P1.0</b>	<b>;faz P1.0 = 1</b>
<b>SJMP LB1</b>	<b>;volta para LB1:</b>

# Exemplo com temporizador no Modo 1

---

# Modo 1



# Exemplo 03

---

## Exemplo 3

---

Usando o Temporizador, gere pelo pino P1.0 uma onda quadrada de 250 kHz

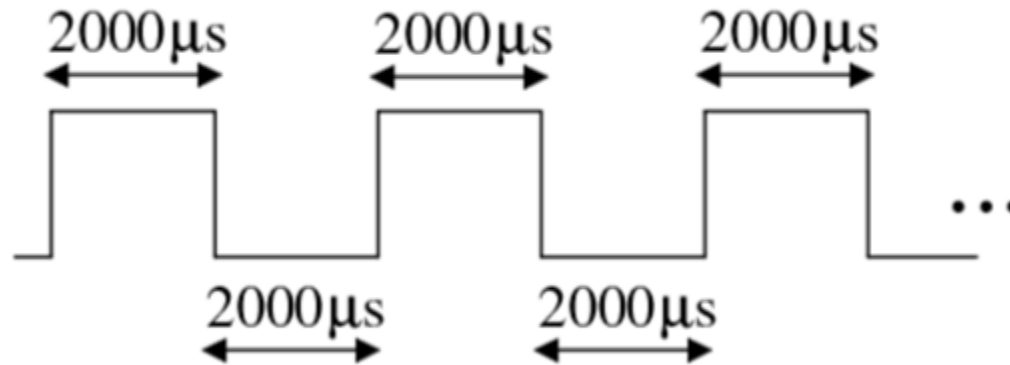


## Exemplo 3

### Solução:

---

O período de uma onda quadrada de 250 kHz é de  $4000\ \mu\text{s}$ , o que significa que o pino P1.0 deve, repetitivamente, ficar  $2000\ \mu\text{s}$  em nível baixo e  $2000\ \mu\text{s}$  em nível alto.



## Exemplo 3

### Solução:

---

Usaremos o modo 1, onde o contador deve ser preparado para sofrer a ultrapassagem após 2000 contagens, e como a contagem é crescente, ele deve ser programado com o valor  $65536 - 2000 = 63536$ .

A contagem sempre começa em 63536 e a ultrapassagem acontece após 2000 contagens ( $63536 + 2000$  que ocorre overflow).

No entanto teremos que converter o valor 63536 para hexadecimal.

63536 é **F830** em hexadecimal.

# Exemplo 3

## Solução:

---

### CONFIG:

```
MOV TMOD,#1
MOV TH0,#0F8h
MOV TL0,#30h
SETB TR0
```

```
;CT0 no modo 1
;valor para F830 - > MSB F8
;valor para F830 - > LSB 30
;liga o CT0
```

### ROT:

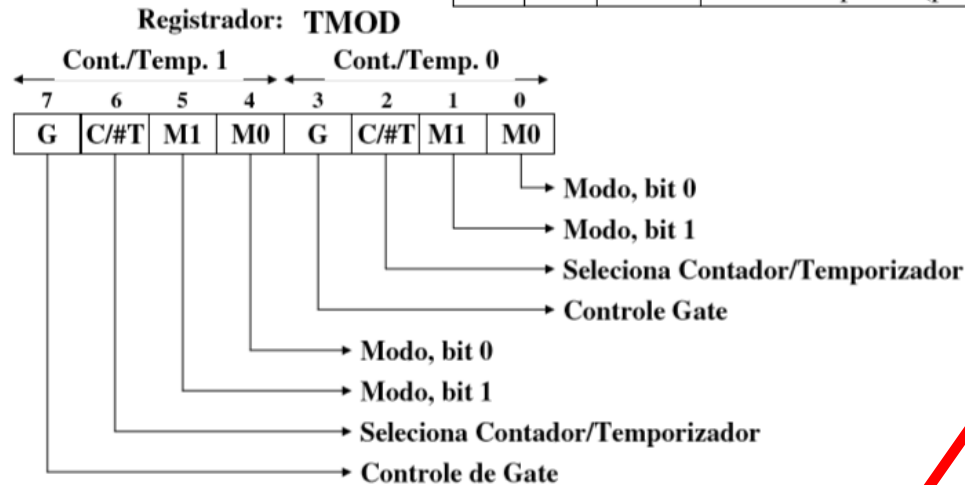
```
JNB TF0,$
MOV TH0,#0F8h
MOV TL0,#30h
CLR TF0
CPL P1.0
SJMP ROT
```

```
;aguarda bit de ultrapassagem
;recarrega com valor para F830 - > MSB F8
;recarrega com valor para F830 - > LSB 30
;apaga bit de ultrapassagem
;complementa P1.0
;fecha o laço
```

# Exemplo 3

## Solução:

M1	M0	Modo	Descrição
0	0	0	THi é CT de 8 bits e TLi é pré-escala de 5 bits
0	1	1	THi e TLi formam CT de 16 bits.
1	0	2	TLi é CT de 8 bits e THi armazena valor de recarga.
1	1	3	TL0 é CT de 8 bits (usando TR0, #INT0 e TF0), TH0 é CT de 8 bits (usando TR1, #INT1 e TF1) TH1 e TL1 parado (pode operar em outros modos)



### CONFIG:

```
MOV TMOD,#1           ;CT0 no modo 1
MOV TH0,#0F8h         ;valor para F830 - > MSB F8
MOV TL0,#30h          ;valor para F830 - > LSB 30
SETB TR0              ;liga o CT0
```

### ROT:

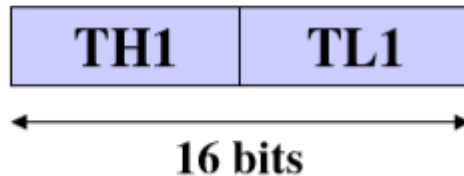
```
JNB TF0,$             ;aguarda bit de ultrapassagem
MOV TH0,#0F8h         ;recarrega com valor para F830 - > MSB F8
MOV TL0,#30h          ;recarrega com valor para F830 - > LSB 30
CLR TF0               ;apaga bit de ultrapassagem
CPL P1.0              ;complementa P1.0
SJMP ROT              ;fecha o laço
```

# Exemplo 3

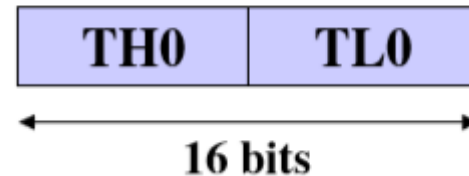
## Solução:

M1	M0	Modo	Descrição
0	0	0	THi é CT de 8 bits e TLi é pré-escala de 5 bits
0	1	1	THi e TLi formam CT de 16 bits.
1	0	2	TLi é CT de 8 bits e THi armazena valor de recarga.
1	1	3	TL0 é CT de 8 bits (usando TR0, #INT0 e TF0), TH0 é CT de 8 bits (usando TR1, #INT1 e TF1) TH1 e TL1 parado (pode operar em outros modos).

Contador/Temporizador 1



Contador/Temporizador 0



### CONFIG:

MOV TMOD,#1	;CT0 no modo 1
MOV TH0,#0F8h	;valor para F830 - > MSB F8
MOV TL0,#30h	;valor para F830 - > LSB 30
SETB TR0	;liga o CT0

### ROT:

JNB TF0,\$	;aguarda bit de ultrapassagem
MOV TH0,#0F8h	;recarrega com valor para F830 - > MSB F8
MOV TL0,#30h	;recarrega com valor para F830 - > LSB 30
CLR TF0	;apaga bit de ultrapassagem
CPL P1.0	;complementa P1.0
SJMP ROT	;fecha o laço

## Exemplo 3

M1	M0	Modo	Descrição
0	0	0	THi é CT de 8 bits e TLi é pré-escala de 5 bits
0	1	1	THi e TLi formam CT de 16 bits.
1	0	2	TLi é CT de 8 bits e THi armazena valor de recarga.
1	1	3	TL0 é CT de 8 bits (usando TR0, #INT0 e TF0), TH0 é CT de 8 bits (usando TR1, #INT1 e TF1) TH1 e TL1 parado (pode operar em outros modos).

## Solução:

No modo 1 não tem recarga automática, portanto você terá que realizar a recarga no TH0 e TL0 após ocorrer o overflow.

### CONFIG:

MOV TMOD,#1	;CT0 no modo 1
MOV TH0,#0F8h	;valor para F830 - > MSB F8
MOV TL0,#30h	;valor para F830 - > LSB 30
SETB TR0	;liga o CT0

### ROT:

JNB TF0,\$	;aguarda bit de ultrapassagem
MOV TH0,#0F8h	;recarrega com valor para F830 - > MSB F8
MOV TL0,#30h	;recarrega com valor para F830 - > LSB 30
CLR TF0	;apaga bit de ultrapassagem
CPL P1.0	;complementa P1.0
SJMP ROT	;fecha o laço

# Exemplo 3

## Solução:

---

### CONFIG:

```
MOV TMOD,#1
MOV TH0,#0F8h
MOV TL0,#30h
SETB TR0
```

```
;CT0 no modo 1
;valor para F830 - > MSB F8
;valor para F830 - > LSB 30
;liga o CT0
```

### ROT:

```
JNB TF0,$
MOV TH0,#0F8h
MOV TL0,#30h
CLR TF0
CPL P1.0
SJMP ROT
```

```
;aguarda bit de ultrapassagem
;recarrega com valor para F830 - > MSB F8
;recarrega com valor para F830 - > LSB 30
;apaga bit de ultrapassagem
;complementa P1.0
;fecha o laço
```

# Bibliografia

---

ZELENOVSKY, R.; MENDONÇA, A. Microcontroladores Programação e Projeto com a Família 8051. MZ Editora, RJ, 2005.

Gimenez, Salvador P. Microcontroladores 8051 - Teoria e Prática, Editora Érica, 2010.