

## Uso do Temporizador (Timer) / Contador

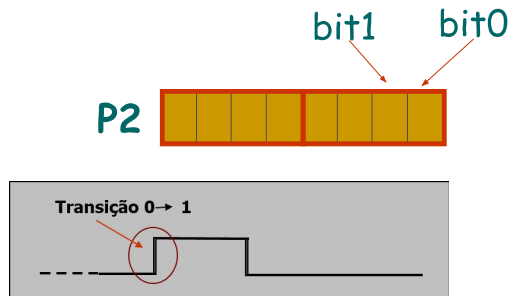
Elimine o loop de software usado para contar o tempo no experimento 2. Use em seu lugar o temporizador para contar tempo. Uma rotina não pode interferir na outra.

Se o bit 0 de P2 mudar de 0 para 1

Atribua os 4 bits menos significativos de P0 aos 4 bits menos significativos de P1

Senão

Espere 1 segundo (use **Timer 0**) e atribua zero aos 4 bits menos significativos de P1

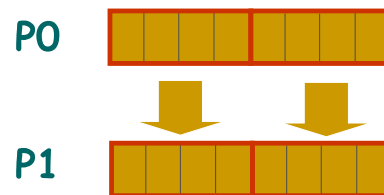


Se o bit 1 de P2 mudar de 0 para 1

Atribua os 4 bits mais significativos de P0 aos 4 bits mais significativos de P1

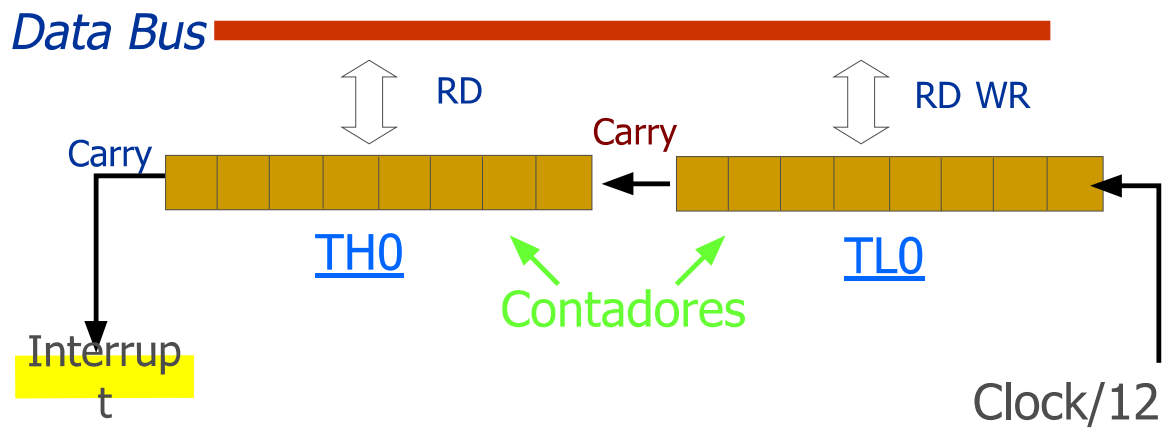
Senão

Espere 1 segundo (use **Timer 0**) e atribua zero aos 4 bits mais significativos de P1



## Uso do Temporizador (Timer) / Contador

- Dispositivo que gera interrupções em intervalos regulares de tempo.  
Ex: Timer do 8051



## Uso do Temporizador (Timer) / Contador

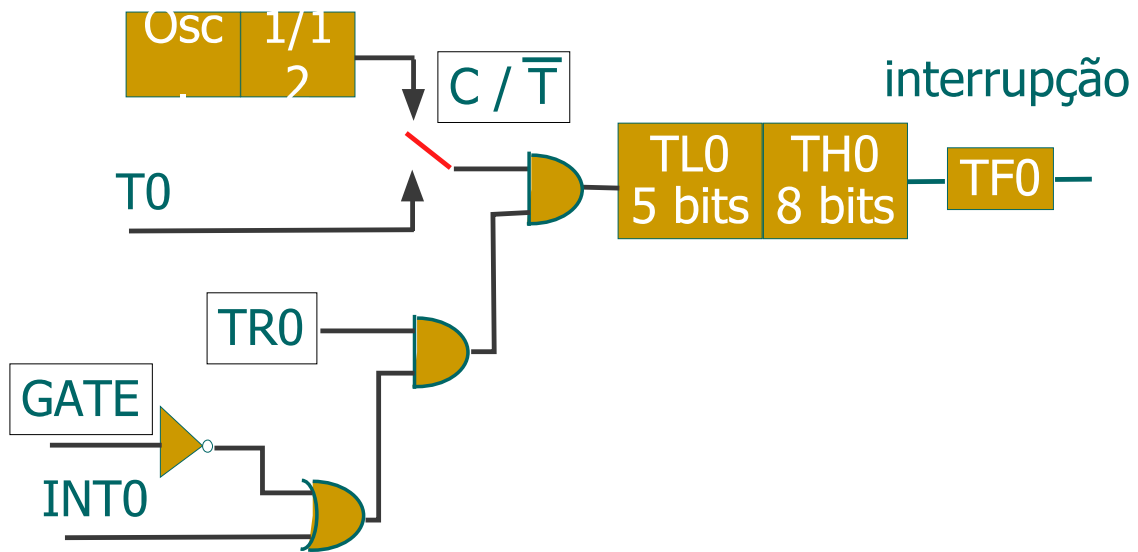
---

- Timer0 e Timer1
- Duas funções: timer e contador
- Frequência máxima como timer
  - Frequência do Clock / 12
- Frequência máxima como contador
  - Frequência do Clock / 24

## Timer 0 no Modo 0

---

- Vários modos de operação:
  - Modo 0:
    - Timer/contador de 13 bits
    - Pode ter interrupção ao chegar em zero.
    - $$\text{Frequência Timer} = \frac{\text{Frequência do Clock}}{(12 * (8192 - \text{ValorInicial}))}$$



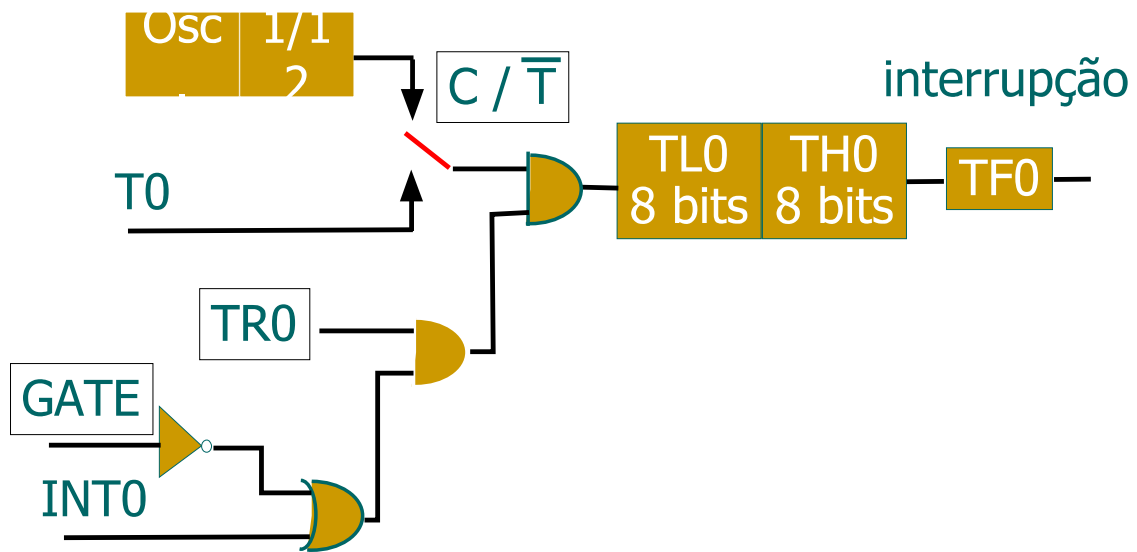
## Timer 0 no Modo 1

---

- Modo 1:

- Timer/contador de 16 bits
- Pode ter interrupção ao chegar em zero.
- Frequência Timer =  $\frac{\text{Frequência do Clock}}{(12 * (65536 - \text{ValorInicial}))}$

# Timer 0 no Modo 1





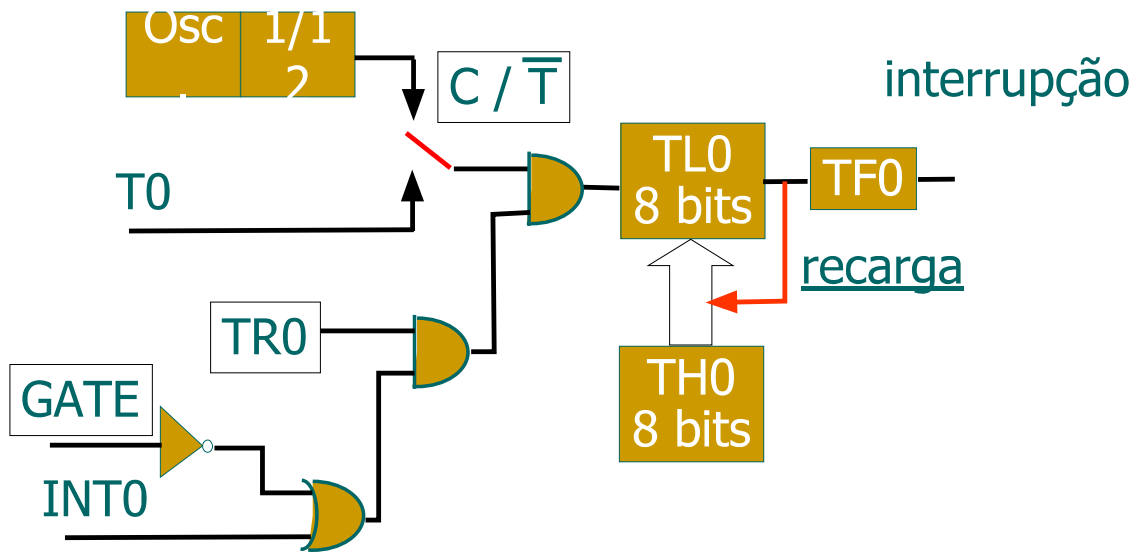
## Timer 0 no Modo 2

---

- Modo 2:

- Timer/contador de 8 bits
- Reload (recarga) automático
- Frequência Timer = 
$$\frac{\text{Frequência do Clock}}{(12 * (256 - \text{Valor THi}))}$$

# Timer 0 no Modo 2



## Timer 0 no Modo 3

---

- Modo 3:
  - TLO funciona como um timer/contador de 8 bits controlado por TR0.
  - TH0 funciona como timer de 8 bits controlado por TR1 e usa TF1 para overflow
  - Timer 1 fica habilitado todo o tempo, mas não pode gerar interrupção. Se Timer 1 também for posto no Modo 3, ele pára.



*TFi* - flag de overflow do Timer i. Se o 8051 for configurado para gerar interrupção quando este flag vai para 1 (um), seu valor volta para 0 (zero) automaticamente após execução da rotina de interrupção.

*TRi* - bit de controle do Timer i

TRi = 0 ⇒ Timer i desligado

= 1 ⇒ Timer i ligado

OBS: os bits de TCON podem ser acessados diretamente

# Timer/Contador: Registradores de Controle

gate	C/T	M1	M0	gate	C/T	M1	M0	TMOD
------	-----	----	----	------	-----	----	----	------

$gate = 0 \Rightarrow$  Timer  $i$  é habilitado se  $TR_i = 1$   
 $= 1 \Rightarrow$  Timer  $i$  é habilitado se  $INT_i = 1$  e  $TR_i = 1$

$C/T = 0 \Rightarrow$  Timer  $i$  funciona como timer  
 $= 1 \Rightarrow$  Timer  $i$  funciona como contador

$M1, M0 = 00 \Rightarrow$  Timer  $i$  funciona no Modo 0  
 $= 01 \Rightarrow$  Timer  $i$  funciona no Modo 1  
 $= 10 \Rightarrow$  Timer  $i$  funciona no Modo 2  
 $= 11 \Rightarrow$  Timer 0 funciona no Modo 3  
 Timer 1 pára.

OBS: os bits de TMOD NÃO podem ser acessados diretamente

## Timer 0 no Modo 1: Programação

---

```
#define FrClk 12000000
#define FreqTimer0_emHz 100
#define TH0_Inicial ((65536-(FrClk/(12*FreqTimer0_emHz))+CORRECAO)>>8)
#define TL0_Inicial ((65536-(FrClk/(12*FreqTimer0_emHz))+CORRECAO)&0xFF)
//Calcule o valor de CORRECAO!!!

main()
{
    timer0_inicializa();
    EA=1; //Habilita o tratamento de interrupções
    ...
}
```

## Timer 0 no Modo 1: Programação

---

```
void timer0_inicializa()
{
    TR0 = 0;                //Desliga Timer0
    TMOD = (TMOD & 0xF0)|0x01; //Timer 0 programado no Modo 1 (16 bits)
    TH0 = TH0_Inicial;      // Programa valor de contagem do Timer0
    TL0 = TL0_Inicial;
    ET0 = 1;                // Habilita interrupcao do timer 0
    TR0 = 1;                // Habilita contagem do timer 0
}

void timer0_int (void) interrupt 1 using 2
{
    TR0 = 0;                // Desliga Timer0
    TL0 += TL0_Inicial;     // Faz recarga da contagem do Timer0
    TH0 += TH0_Inicial + (unsigned char) CY;
    TR0 = 1;                // Habilita contagem do timer 0
    ...
}
```