

# **Mestrado Integrado em Engenharia de Comunicações**

**Unidades**  
**Temporizadoras ou**  
**Contadoras de eventos**

Microprocessadores I  
2º Ano – A11

# Directivas

Categoria	Directiva	Sintaxe			Função
Controlo do estado	<b>ORG</b>		<b>ORG</b>	expressão	Especifica um valor para contador de localização do segmento activo
	<b>END</b>		<b>END</b>		Indica ao <i>assembler</i> o fim do programa fonte
	<b>USING</b>		<b>USING</b>	expressão	Indica ao <i>assembler</i> o banco de registo usado no código que vem a seguir à directiva. Repare que a comutação do banco de registo deve ser efectuada usando apenas instruções do 8051
Definição de símbolos	<b>SEGMENT</b>	Símbolo	<b>SEGMENT</b>	tipo_de_segmento	Declara um símbolo como sendo um segmento <i>relocatable</i> de um dado tipo. Para começar a usar o segmento, deve-se usar a directiva RSEG
	<b>EQU</b>	Símbolo	<b>EQU</b>	expressão	Atribui um valor a um símbolo
	<b>SET</b>	Símbolo	<b>SET</b>	expressão	Igual ao EQU, exceptuando o facto de permitir a redefinição o símbolo
	<b>DATA</b>	Símbolo	<b>DATA</b>	expressão	Atribui ao símbolo um endereço directo da RAM interna
	<b>IDATA</b>	Símbolo	<b>IDATA</b>	expressão	Atribui um endereço da RAM interna indirectamente endereçável ao símbolo
	<b>XDATA</b>	Símbolo	<b>XDATA</b>	expressão	Atribui ao símbolo um endereço da memória externa
	<b>BIT</b>	Símbolo	<b>BIT</b>	expressão	Atribui um endereço directo da área de memória endereçável ao bit a um símbolo
	<b>CODE</b>	Símbolo	<b>CODE</b>	expressão	Atribui um endereço da memória de código ao símbolo

# Directivas

Categoria	Directiva	Síntaxe	Função
Inicialização e reserva de armazenamento	<b>DS</b>	[LABEL:] <b>DS</b> expressão	Reserva espaços em múltiplos de <i>bytes</i> . Não pode ser utilizado com segmento do tipo <b>BIT</b> . O valor da expressão deve ser conhecida pelo <i>assembler</i>
	<b>DBIT</b>	[LABEL:] <b>DBIT</b> expressão	Reserva espaços em múltiplos de bits. O valor da expressão deve ser conhecida pelo <i>assembler</i>
	<b>DB/DW</b>	[LABEL:] <b>DB/DW</b> expressão	Inicializa a memória de código com valores do tipo byte/word
<i>Program linkage</i>	<b>PUBLIC</b>	<b>PUBLIC</b> Símbolo [, símbolo ] [...]	Define uma lista de símbolos que tornam visíveis e utilizáveis a partir de outros módulos
	<b>EXTRN</b>	<b>EXTRN</b> Tipo_segmento(símbolo [,símbolo] [...], ...)	Informa o <i>assembler</i> da lista de símbolos definidos noutros módulos e que vão ser utilizados neste. O tipo de segmento pode ser <b>CODE</b> , <b>DATA</b> , <b>XDATA</b> , <b>IDATA</b> , <b>BIT</b> e um especial designado por <b>NUMBER</b> que especifica um símbolo definido por <b>EQU</b>
	<b>NAME</b>	<b>NAME</b> Nome_do_módulo	
Seleccção de Segmentos	<b>RSEG</b>	<b>RSEG</b> Nome_do_segmento	Ao encontrar uma directiva de selecção de segmento, o <i>assembler</i> direcciona o código
	<b>CSEG</b>	<b>CSEG</b> [ AT endereço ]	ou dado que lhe segue para o segmento
	<b>...</b>	<b>DSEG</b> [ AT endereço ]	seleccionado até que seja seleccionado um
	<b>XSEG</b>	<b>XSEG</b> [ AT endereço ]	outro segmento

# Exemplos

<pre> 01 #include &lt;89C51Rx2.inc&gt; 02 03 VAR1    DATA    30H 04 VAR2    DATA    31H 05 VAR3    DATA    32H 06 </pre>	<pre> 07 DSEG    AT        40H 08 VAR4:    DS        1 09 VAR5:    DS        1 10 VAR6:    DS        1 11 12 VARI1    IDATA     80H 13 VARI2    IDATA     81H 14 15 VAREXT   XDATA     100H </pre>	<pre> 21 MOV      A,#55H 22 MOV      VAR1,A 23 INC      A 24 MOV      VAR2,A 25 CPL      A 26 MOV      VAR6,A 27 MOV      R0,#VARI1 28 MOV      @R0,#55H </pre>
--	--	---

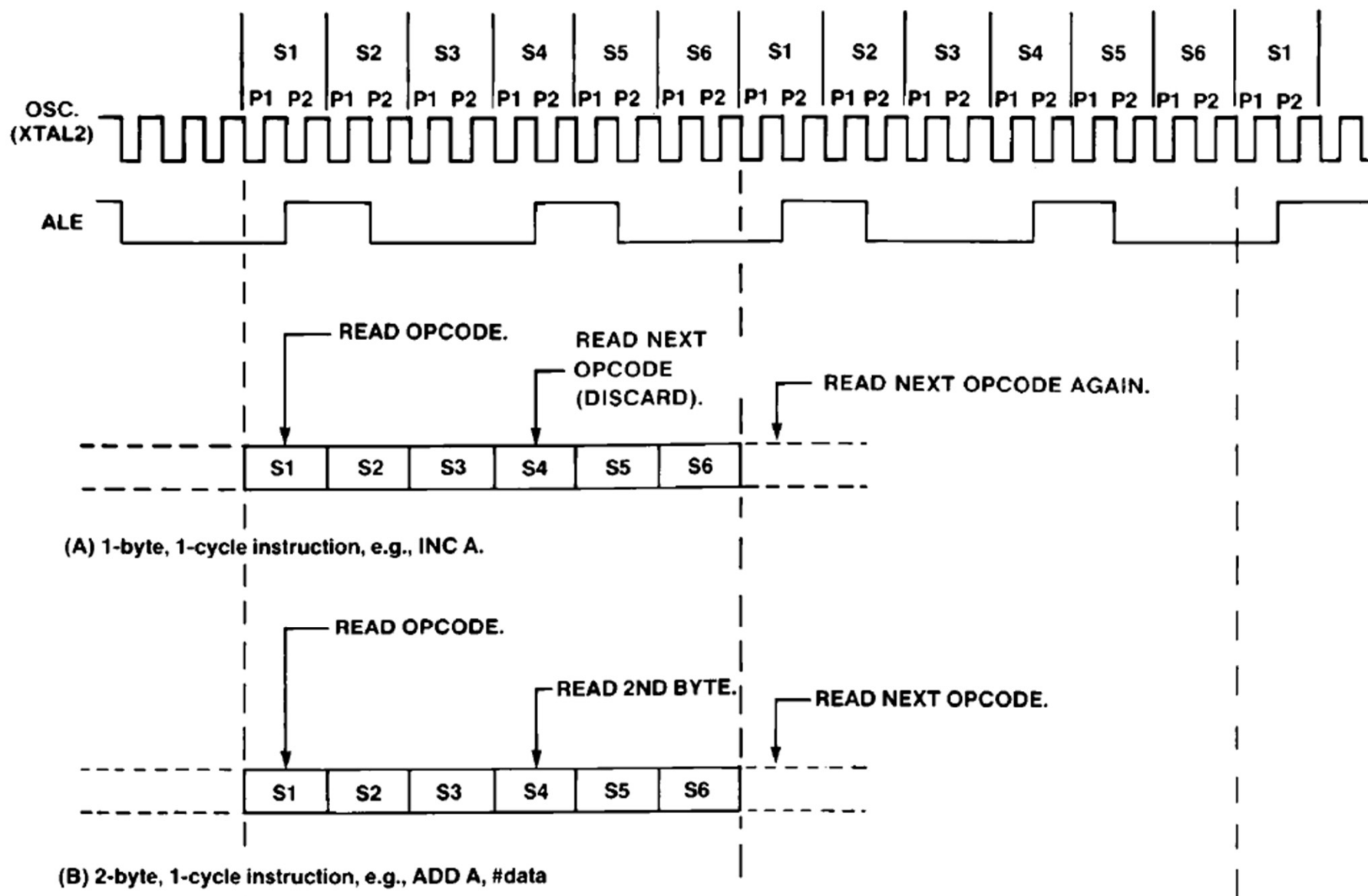
Watch 1

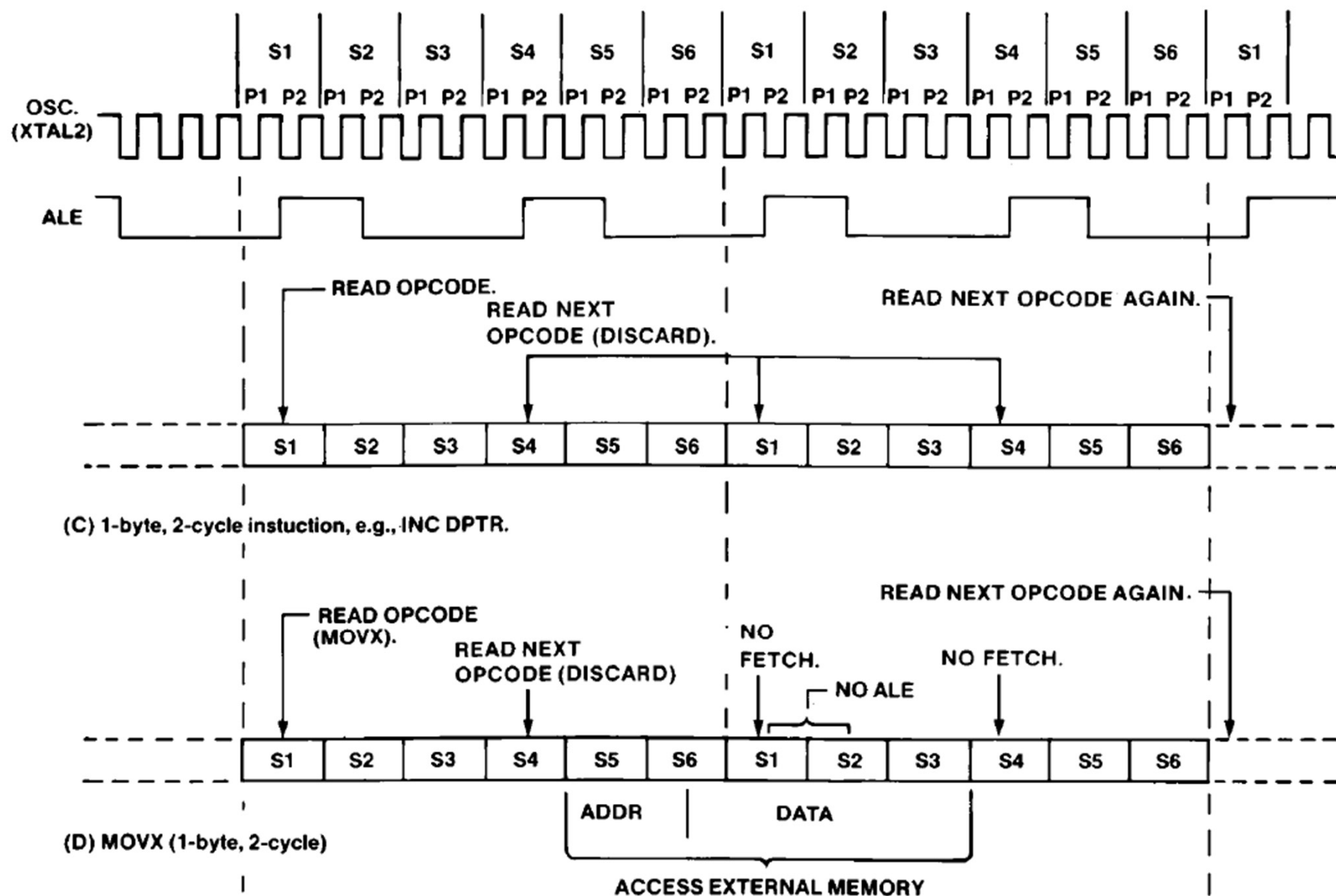
Name
VARI1
VAREXT
<double-click or F2 to add>

<pre> 01 #include &lt;89C51Rx2.inc&gt; 02 03 BSEG     AT        0H 04 BITON:   DBIT      1 05 LEDVERDE: DBIT      1 06 FLAG3:   DBIT      1 07 MOTORESQ: DBIT      1 08 09 XSEG     AT        300H 10 VAR1:    DS        1 11 </pre>	<pre> 12 CSEG     AT        0H 13 JMP      MAIN 14 CSEG     AT        50H 15 MAIN: 16 MOV      A,#55H 17 MOV      20H,A 18 SETB     LEDVERDE 19 CLR      FLAG3 20 MOV      A,20H </pre>
--	---

Watch 1

Name
LEDVERDE
BITON
FLAG3
<double-click or F2 to add>





# Rotinas de atraso (delay)

**DELAY1:** **MOV** **R2,#X**  
**DJNZ** **R2,\$**  
**RET**

O nº de ciclos máquina desta rotina é:  
 **$NC=1+2*X+2=3+2*X$**   
 **$X=1 \Rightarrow NC=5$        $X=0 \Rightarrow NC=515$**   
 **$delay\ (seg)=12*NC/(frequência\ cristal)$**

**DELAY2:** **MOV** **R3,#Y**  
**D2LOOP:** **MOV** **R2,#X**  
**DJNZ** **R2,\$**  
**DJNZ** **R3,D2LOOP**  
**RET**

O nº de ciclos máquina desta rotina é:  
 **$NC=(3+2*X)*Y+3$**

**DELAY3:** **MOV** **R4,#Z**  
**D3L1:** **MOV** **R3,#Y**  
**D3L2:** **MOV** **R2,#X**  
**DJNZ** **R2,\$**  
**DJNZ** **R3,D3L2**  
**DJNZ** **R4,D3L1**  
**RET**

O nº de ciclos máquina desta rotina é:  
 **$NC=((3+2*X)*Y+3)*Z+3$**

# *Rotinas de atraso (delay)*

- **Delay por software:**
  - O tempo de espera é estabelecido pelo número de ciclos máquina necessários para executar a rotina de *delay*. O microprocessador fica bloqueado, ou seja, durante a execução do *delay* não pode executar outro código;
- **Difíceis de controlar:**
  - Para além de ocuparem registos, o valor a colocar em cada registo não é “simples” de obter. Muitas vezes opta-se por implementar uma rotina de *delay* fixo (ex:1000µs) e invocá-la várias vezes;
- **Dependem:**
  - Do número de registos e dos seus valores, do nº de ciclos máquina necessários à execução das instruções e do oscilador utilizado.



## *Rotinas de atraso (delay)*

- Suponham que pretendemos gerar uma onda quadrada no pino P1.0. Qual a maior frequência possível e qual o *duty-cycle* dessa onda?

<b>ONDA:</b>	SETB	P1.1	;NC=1 $\Rightarrow$ 1 $\mu$ s	
	CLR	P1.1	;NC=1 $\Rightarrow$ 1 $\mu$ s	T=4 $\mu$ s $\Rightarrow$ f=250KHz
	SJMP	ONDA	;NC=2 $\Rightarrow$ 2 $\mu$ s	D=t <sub>on</sub> /T*100=1 $\mu$ s/4 $\mu$ s=25%

- Altere o código de modo a garantir um *duty-cycle* de 50%. Qual a frequência da onda quadrada?
- Faça um rotina que permita gerar uma onda quadrada de 20KHz com um *duty-cycle* de 50%.

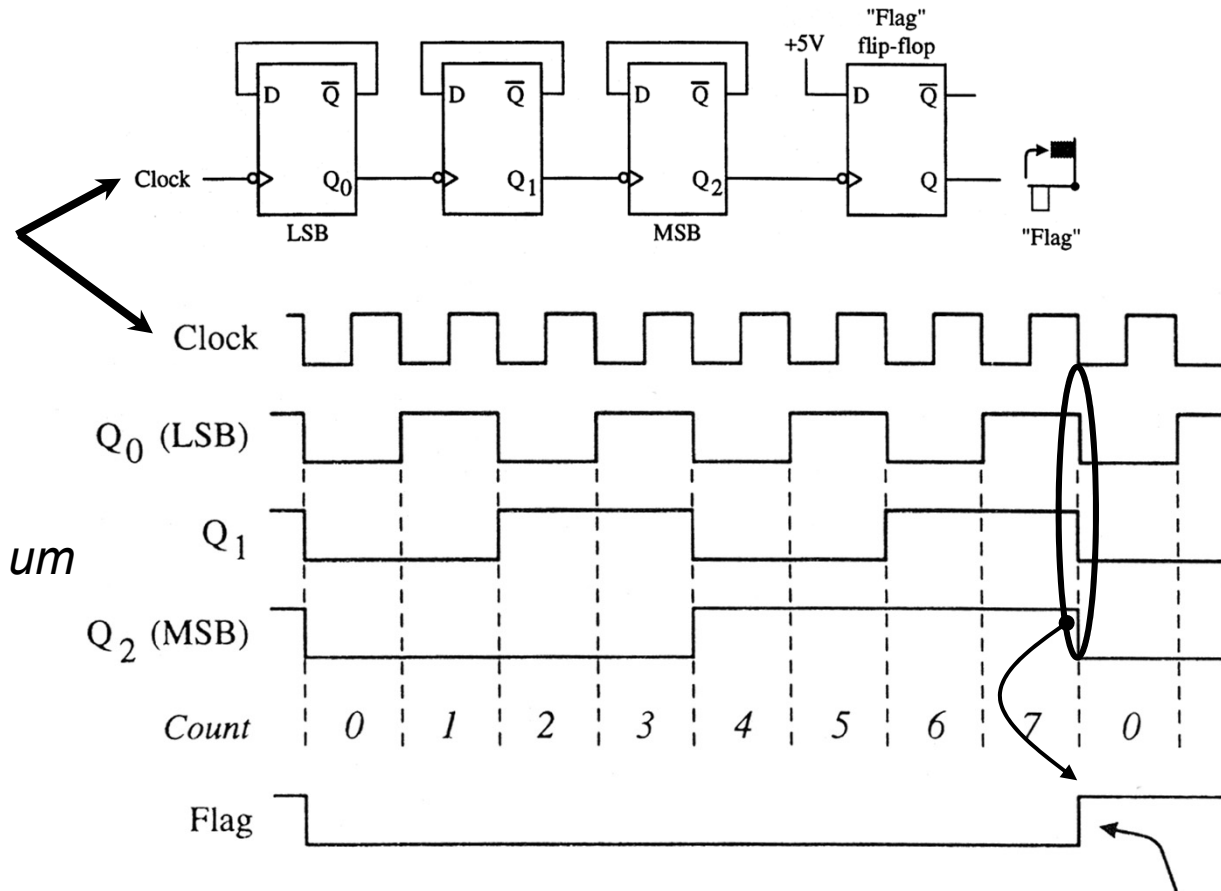
# Timers/Counters

## Como se divide um sinal de relógio?

O sinal de clock seria obtido através do relógio do microcontrolador – **temporizador**

ou

através de um sinal de relógio externo ligado a um pino de E/S do microcontrolador - **contador de eventos**



# *Timers/Counters*

- *O 8051 tem duas unidades de temporização e contagem:*
  - *Timer 0 e Timer 1;*
  - Os modelos da família 8052 têm mais uma unidade: *Timer 2*.
- *Os timers podem operar como temporizadores ou como contadores de eventos externos ao microcontrolador:*
  - Quando opera como temporizador, os registos do *timer*, são incrementados a cada ciclo máquina (utiliza o relógio do CPU), ou seja a taxa de contagem é 1/12 da frequência do relógio;
  - No modo de operação de contagem de eventos externos, os registos são incrementados a cada transição de 1 para 0 na entrada externa do *timer*.

# Timers/Counters

- *No modo de contagem, os registos do timer são incrementados sempre que há uma transição de 1-para-0 no respectivo pino de entrada (T0, T1 ou T2).*
  - O pino de entrada é amostrado durante o estado **S5P2** do ciclo de instrução.
  - O incremento é feito quando num ciclo de instrução a entrada estiver a '1' e no ciclo seguinte estiver a '0'.
    - *O registo é actualizado durante o estado **S3P1** do estado a seguir a detecção.*
    - *Uma vez que são necessários dois ciclos de instrução, a maior taxa de contagem permitida é de 1/24 da frequência do relógio.*

# *Timers/Counters*

- *Os timers 0 e 1 permitem quatro modos de funcionamento:*
  - *Modo 0: temporizador/contador de 13 bits.*
  - *Modo 1: temporizador/contador de 16 bits.*
  - *Modo 2: temporizador/contador de 8 bits com auto-reload.*
  - *Modo 3: duplo temporizador/contador de 8 bits.*

# Timers/Counters

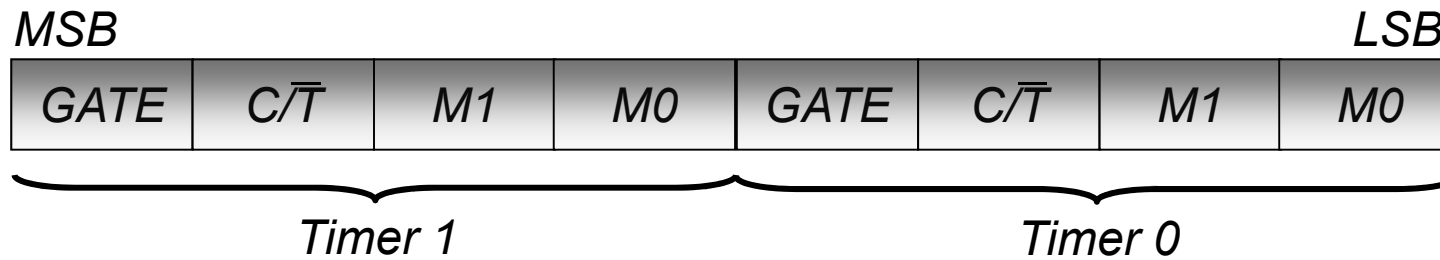
- *Registos usados na programação dos timers 0 e 1:*
  - TMOD: Permite programar os modos de funcionamento dos timers.
  - TCON: Permite controlar a activação e verificar o estado dos timers.
  - THx/TLx: Registo de 16 bits (THx: MSB, TLx: LSB).

*Contêm os valores de contagem de eventos ou para a temporização de um intervalo de tempo*
  - IE: Permite controlar a geração da interrupção associada aos timers]

(TH1:8Dh      TH0:8Ch      TL1:8Bh      TL0:8Ah)  $\Rightarrow$  **SFR**

# Timers/Counters

- *Registo: **TMOD** (89h)*



*GATE:* Quando activado ('1'), o timer x só é habilitado quando /INTx está a '1' e o pino de controlo TRx está activado ('1'). Quando desactivado, o timer x é habilitado quando TRx está activado ('1').

*C/ $\bar{T}$ :* Selecciona o modos de funcionamento que podem ser como temporizador ou como contador (contagem feita através do sinal no pino Tx).

*M0, M1:* Selecciona os modos de operação.

# Timers/Counters

- *Registro: **TMOD** (89h)*

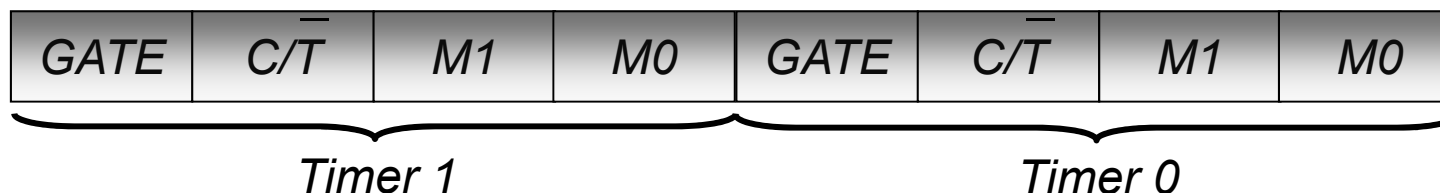
*M1, M0: 0 0 Modo 0: Temporizador/Contador X de 13 bits*

0 1 Modo 1: Temporizador/Contador X de 16 bits

1 0    *Modo 2: Temporizador/Contador X de 8 bits  
com auto-carregamento*

1 1      *Modo 3: Timer 0 – TL0 é um temporizador/contador de 8 bits controlado pelos bits de controlo do timer 0. TH0 é um temporizador/contador de 8 bits controlado pelos bits de controlo do timer 1.*

**1 1** *Modo 3: Timer 1 – Está deshabilitado.*





# Timers/Counters

- Registo: **TCON\*** (88h)

MSB				LSB			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

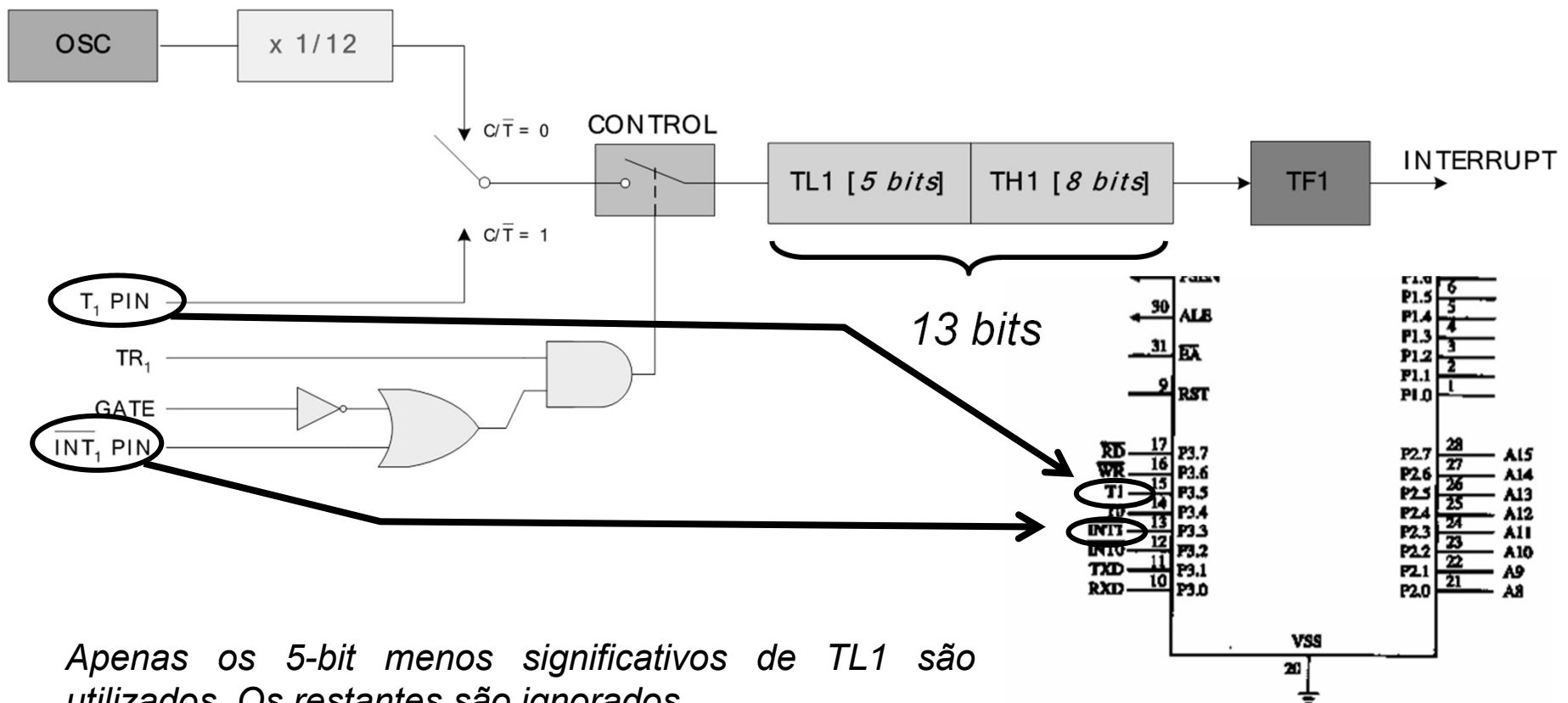
**TFx:** *Timer overflow flag. É colocada a '1' pelo hardware, quando há overflow na contagem do timer. É limpa automaticamente pelo hardware, quando a rotina de serviço a interrupção é chamada. Caso não se use as interrupções, deve ser limpa por software.*

**TRx:** *Bit de controlo do timer x. Activado/limpo por software para habilitar/desabilitar o timer x.*

**[IEx, ITx:** *Gestão das interrupções externas 0 e 1]*

# Timers/Counters

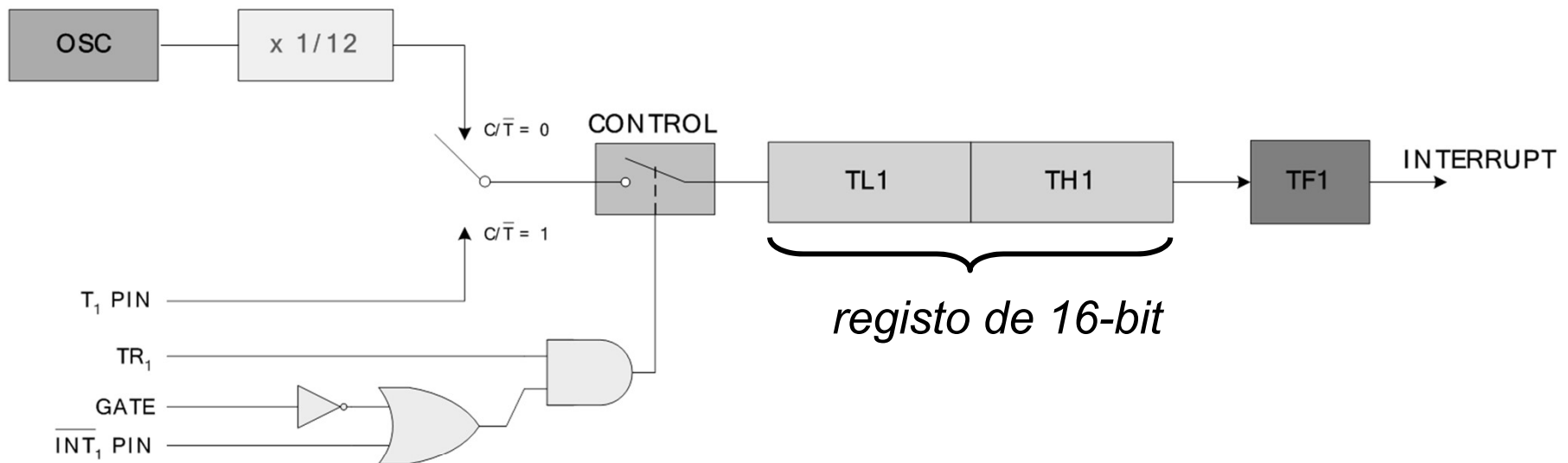
- Timer: funcionamento no modo 0*



Apenas os 5-bit menos significativos de TL1 são utilizados. Os restantes são ignorados.  
 TL1 funciona como um pré-divisor por 32.

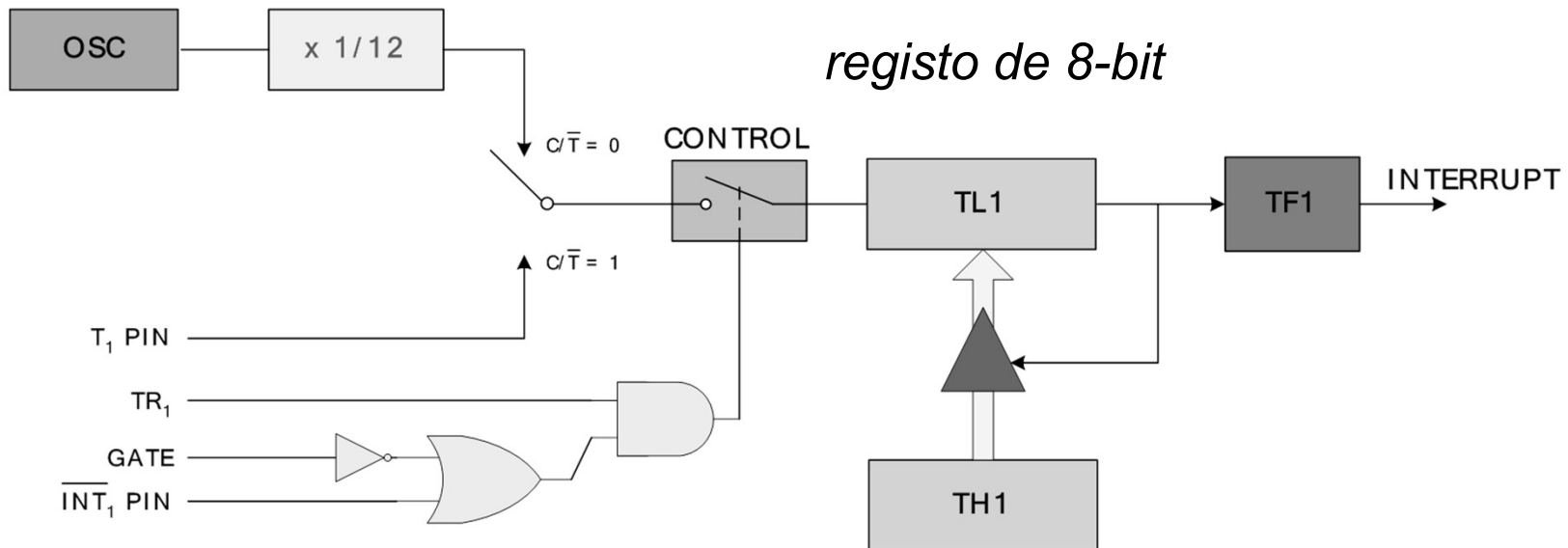
# Timers/Counters

- *Timer: funcionamento no modo 1*



# Timers/Counters

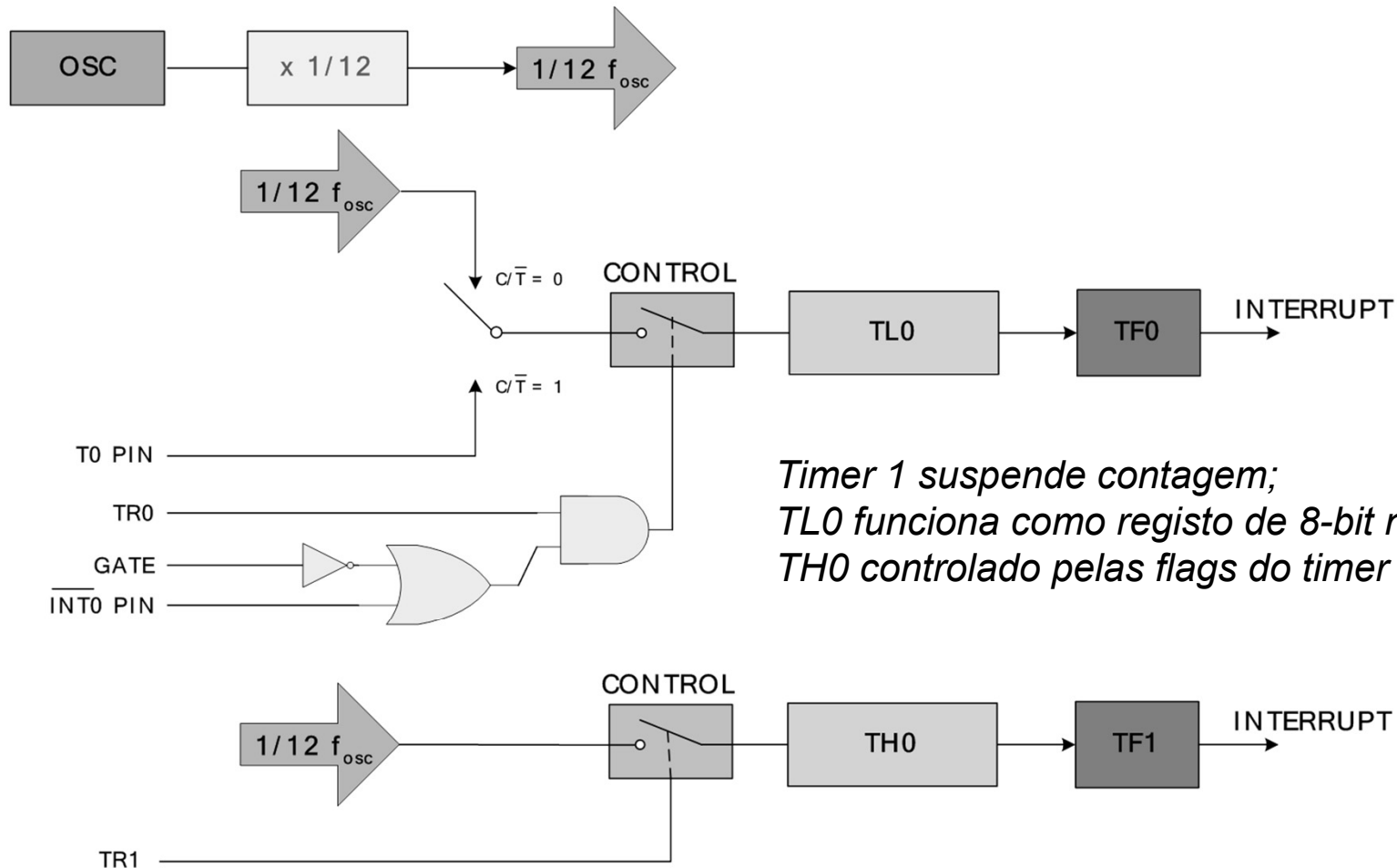
- *Timer: funcionamento no modo 2*



*Sempre que ocorre o overflow do registo TLx, este é recarregado com o valor de THx.*

# Timers/Counters

- *Timer: funcionamento no modo 3*



# *Timers/Counters*

- *O timer 2 é um timer/counter de 16-bit encontrado apenas na família 8052;*
- *Este permite três modos de funcionamento:*
  - **Modo 0:** *temporizador/contador de 16 bits com auto-carregamento;*
  - **Modo 1:** *modo de captura de 16 bits;*
  - **Modo 2:** *gerador de baud rate para as comunicações série.*

# Timers/Counters

- *Registos usados na programação do timer 2:*
  - T2CON: Permite controlar a activação e verificar o estado dos timers – P89C51 ainda tem o T2MOD;
  - TH2/TL2: Registo de 16 bits (TH2: MSB, TL2: LSB);
  - RCAP2H/RCAP2L: Registo de 16 bits usado na captura e recarregamento;
  - [IE: Permite controlar a geração da interrupção associada aos timers]

(**T2CON**\*:0C8h;**T2MOD**:0C9h;**TH2**:0CDh;**TL2**:0CCh;**RCAP2H**:0CBh;**RCAP2L**:0CAh) ⇒ **SFR**

# Timers/Counters

- *Registo: **T2CON** \*(0C8h)*

MSB				LSB			
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

*TF2: Quando activa indica o overflow do timer 2. Esta flag deve ser limpa por software;*

*EXF2: Activada quando a captura ou carregamento ocorre devido uma transição negativa de T2EX (pino externo) e EXEN2= 1. Se a interrupção estiver habilitada, ocorrerá uma interrupção e EXF2 deve ser limpa por software;*

*RCLK: Quando habilitado indica que o relógio da comunicação série durante a recepção deve ser obtido do timer 2;*

*TCLK: Idem, mas para a transmissão.*



# Timers/Counters

- Registo: **T2CON\*** (0C8h)

MSB				LSB			
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2

*EXEN2: Habilita a entrada externa. Quando activada permite que a captura ou o carregamento dependam de uma entrada externa*

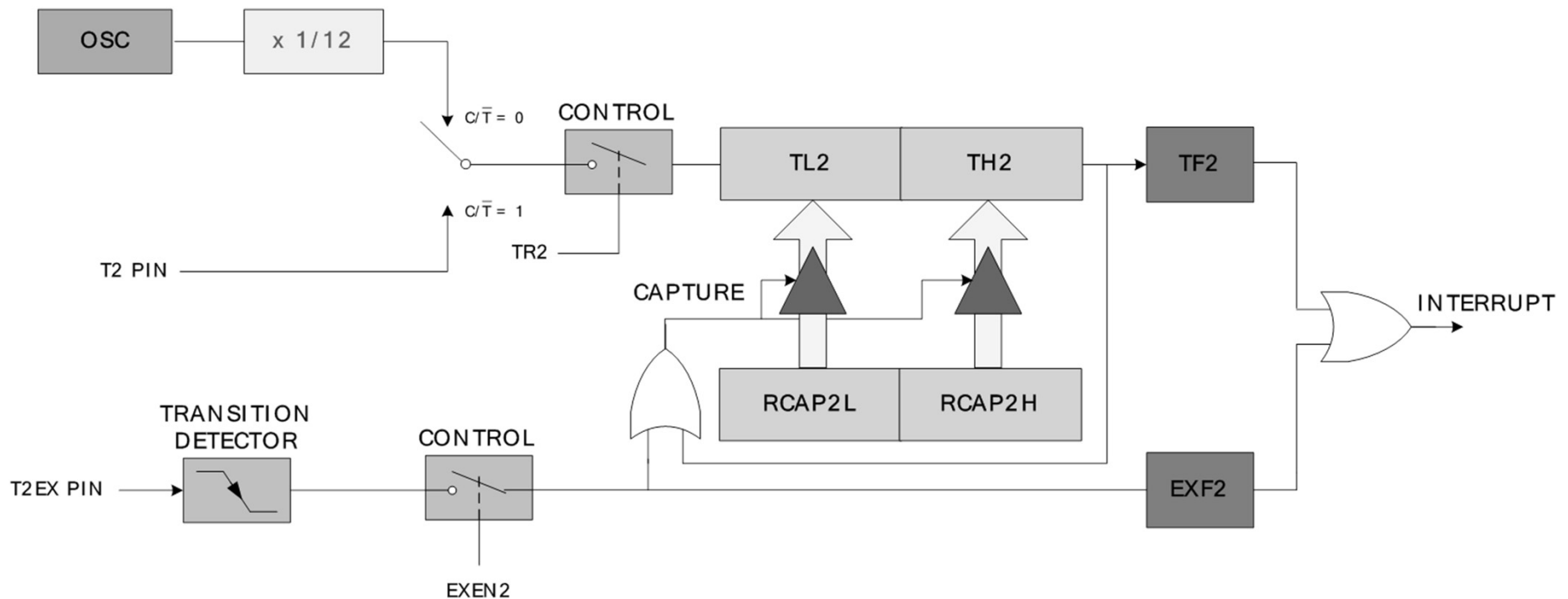
*TR2: Arranque/paragem do timer 2*

*C/ $\bar{T}$ 2: Selecção do modo temporizador (0) ou contador (1)*

*CP/ $\bar{R}$ L2: Selecção entre o modo de captura ou de carregamento*

# Timers/Counters

- Timer 2: Modo 0 / Auto-reload*

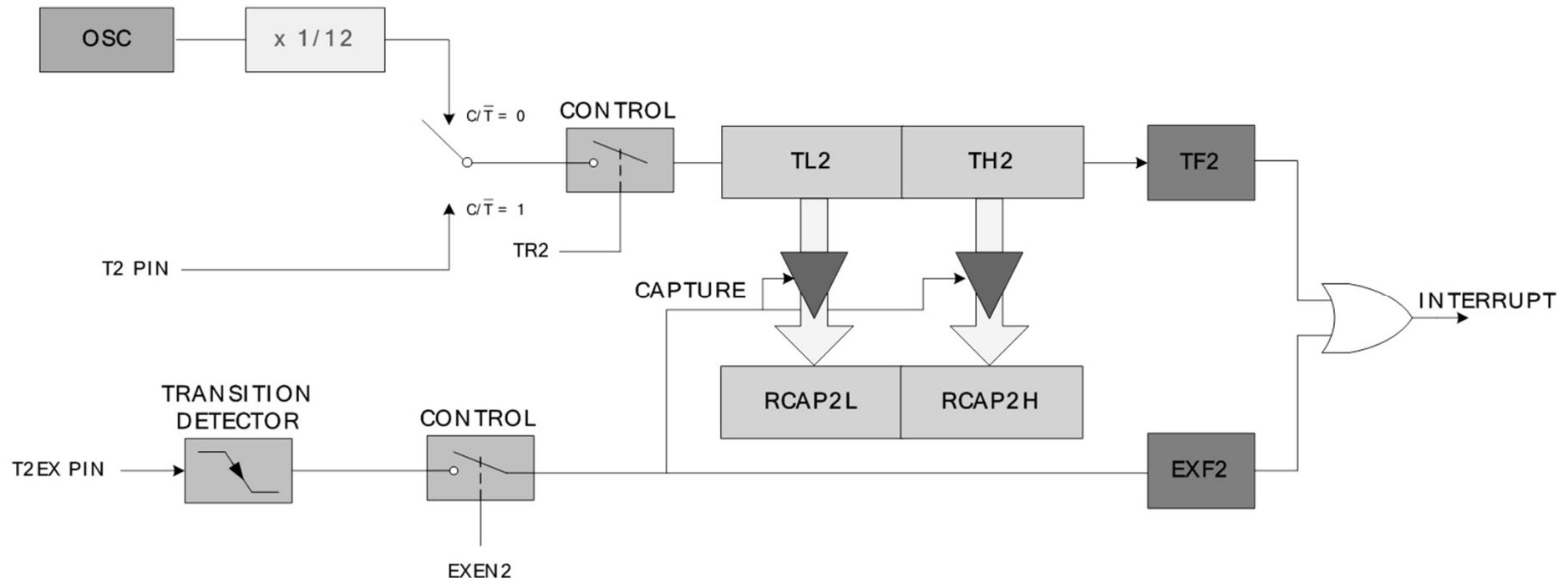


**CP/RL2=0:** timer de 16-bit com auto-reload de RCAP2L e RCAP2H. Colocando EXEN2 a 1 o reload também ocorre numa transição de 1-para-0 no pino T2EX.

**TF2 deve ser limpo por software antes de nova activação**

# Timers/Counters

- *Timer 2: Modo 1 /Captura*



Neste modo, **CP/RL2=1** e **EXEN2=1**.  
Uma transição em T2EX captura o valor nos registos TH2 e TL2 para os registos RCAP2H e RCAP2L

TF2 sinaliza overflow do timer e deve ser limpo por software antes de nova activação

# Timers/Counters

- *Exemplo*

*Escreva um programa que gere uma onda quadrada com 1 KHz de frequência no pino P1.0, usando o timer 0.*

## Análise:

1. *Repare que o período de uma onda quadrada de 1KHz é de 1000  $\mu$ s, sendo que o tempo em alto é igual ao do tempo em baixo 500  $\mu$ s;*
2. *Como este intervalo de 500  $\mu$ s é superior a 256  $\mu$ s torna-se impossível usar o timer no modo auto-reload porque neste modo funciona como temporizador de 8 bits;*
3. *Os timers contam no sentido crescente e activam a flag de overflow na transição FFFFh-para-0000h. Assim sendo, para um cristal de 12MHz, o timer tem de ser inicializado com -500=FE0Ch-até-0000h*
  - i. *O valor a carregar em TL0/TH0 seria -500=FE0Ch, isto é: (TL0)=0Ch e (TH0)=0FEh*

*Timer0 configurado no modo 1 (temporizador de 16 bits) sendo o auto-reload efectuado após cada overflow pelo software a implementar*

# Timers/Counters

- *Exemplo*

*Qual é o problema com esta solução?*

```
MOV TMOD, #01H      ; 16-bit timer mode.
LOOP: MOV TH0, #0FEH  ; -500 (MSB).
      MOV TL0, #0C0H  ; -500 (LSB).
      SETB TR0        ; início da contagem
      JNB TF0, $       ; esperar pelo fim da contagem
      CLR TR0         ; parar a contagem --- não é necessário
      CLR TF0         ; limpar a flag de overflow --- obrigatório
      CPL P1.0        ; comutar o bit 0 do porto P1
      SJMP LOOP
END
```

*Haverá sempre um desvio à frequência pretendida devido às instruções de re-inicialização do timer após cada overflow*