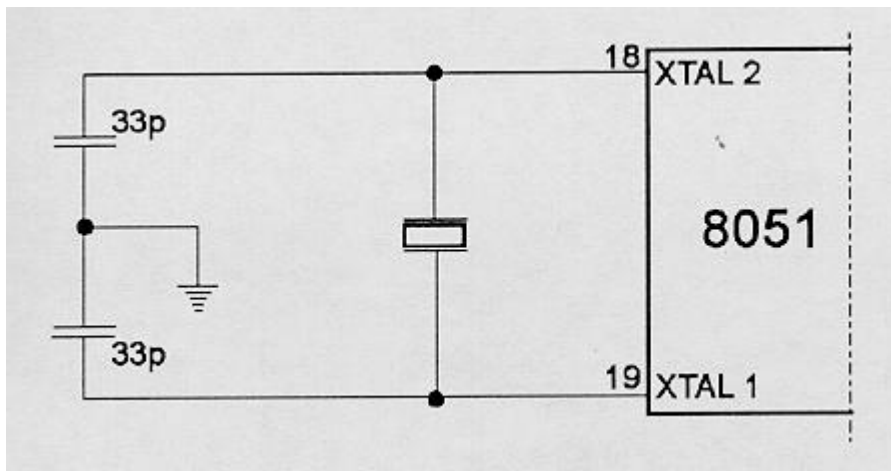
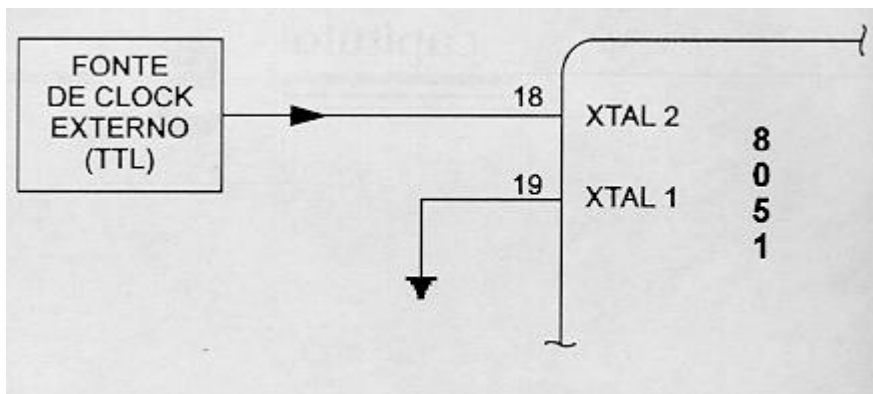


2.4 Temporização da CPU

Todos os Microcontroladores da família MCS-51 têm um oscilador interno. Para uso deste deve-se conectar um cristal entre os pinos Xtal1 e Xtal2.



Pode-se também utilizar um oscilador externo conforme mostra a figura abaixo:



O oscilador (clock) define a seqüência de estados que formam um ciclo de máquina .

- **Ciclos de Máquina**

Um ciclo de máquina (M) consiste de uma seqüência de 6 estados (S1 a S6). Cada estado é formado por 2 períodos de clock (P1 e P2). Logo :

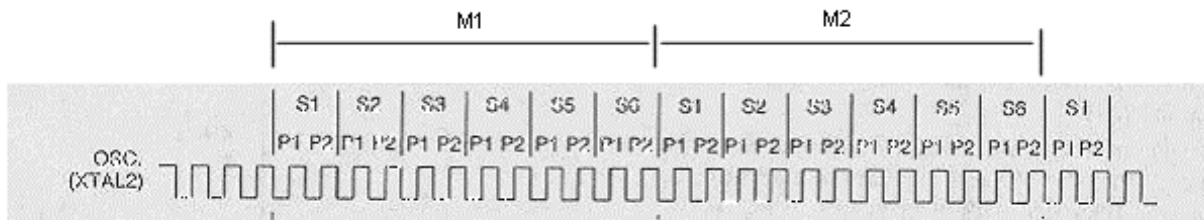
1 ciclo de máquina (M) = 12 períodos de clock (P)

Se o cristal é de 12 Mhz:

$$P = \frac{1}{12 \cdot 10^6} \quad (\text{Período do clock})$$

assim, cada Ciclo de Máquina (M):

$$M = 12 \cdot P = 12 \cdot \frac{1}{12 \cdot 10^6} = 1 \mu s$$



As instruções dos microcontroladores da família MCS-51 utilizam 12 ou 24 períodos de clock, com exceção das instruções **MUL AB** e **DIV AB** que utilizam 48 períodos .

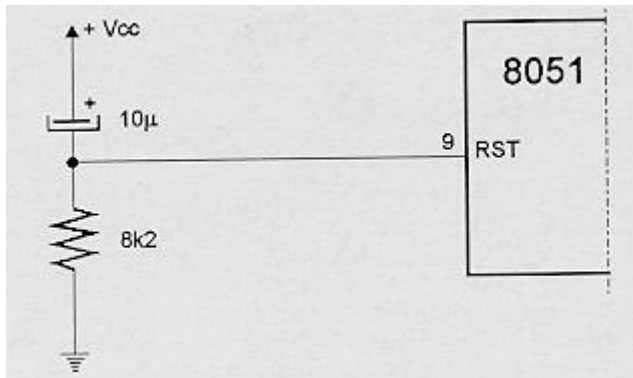
Exemplo : Com cristal de 12 Mhz .

```

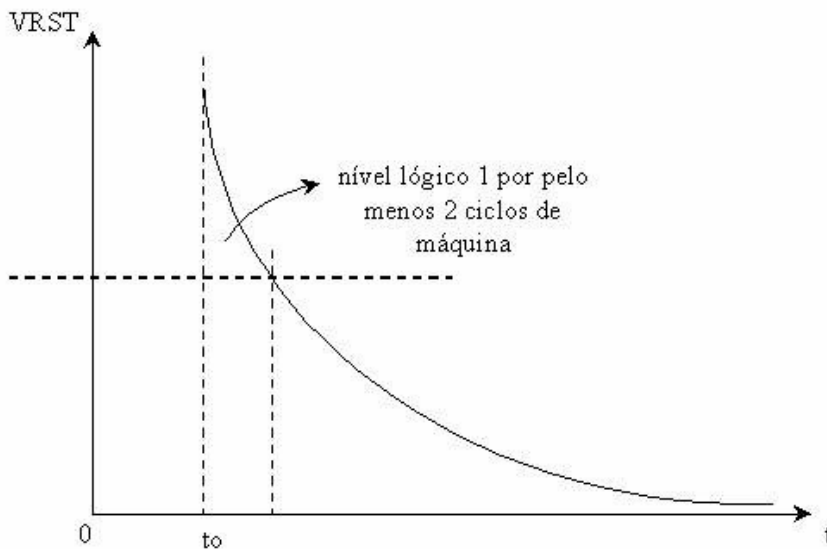
mov R0,a      12 P  1 us
mov R0,#3Fh  24 P  2 us
setb P0.1    12 P  1 us
Djnz R1,loop 24 P  2 us

```

- **Reset da CPU**



- **Reset de Power-up (reset automático)**



Obs: Uma chave pode ser colocada em paralelo com o capacitor para que se possa realizar o reset manual.

2.5 Estrutura de interrupção

Interrupção é uma característica de um computador que permite ao mesmo parar a execução de um determinado programa e passar a executar uma sub-rotina, localizada em um endereço pré-determinado da memória. A sub-rotina a ser executada é denominada de **Sub-rotina de Atendimento de Interrupção**. Ao terminar a execução desta sub-rotina o controle volta para o programa inicial no endereço imediatamente abaixo do ponto onde foi interrompido.

O Microcontrolador 8051 possui 5 fontes de Interrupção :

2 interrupções externas
2 interrupções dos temporizadores
1 interrupção da porta serial

As 5 fontes de Interrupção são:

1. Timer 0 Overflow.
2. Timer 1 Overflow.
3. Reception/Transmission of Serial Character.
4. External Event 0.
5. External Event 1.

Obs : outros membros da família MCS-51 podem ter interrupções adicionais

- **Endereço das interrupções:**

Cada interrupção causa um salto para um endereço fixo na memória de programa a partir do endereço 0003h . Existem 8 bytes de intervalo entre dois endereços consecutivos, para que possa ser colocada uma sub-rotina de atendimento de interrupção ou um JUMP caso não haja espaço suficiente. Se as interrupções não forem usadas os endereços podem ser usados para o programa principal.

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

- **Habilitação das interrupções:**

Cada interrupção pode ser individualmente habilitada setando-se um bit no SFR chamado Registrador IE (Interrupt Enable).

Registrador IE: (endereçável a Bit)

(MSB)								(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0	

Bit = 1 --> Habilita a Interrupção
Bit = 0 --> Desabilita a Interrupção

Símbolo	Posição	Função
EA	IE.7	Desabilita todas as Interrupções. Se EA=0, nenhuma Interrupção será reconhecida. Se EA=1, cada fonte de Interrupção será habilitada ou desabilitada individualmente.
—	IE.6	
ET2	IE.5	Bit de habilitação do Timer 2
ES	IE.4	Bit de habilitação da Porta Serial
ET1	IE.3	Bit de habilitação do Timer 1
EX1	IE.2	Bit de habilitação da Interrupção Externa 1
ET0	IE.1	Bit de habilitação do Timer 0
EX0	IE.0	Bit de habilitação da Interrupção Externa 0

Obs: Quando ocorre um sinal de Interrupção o programa desvia para a Sub-rotina de Atendimento de Interrupção. Se ocorrer outro sinal de Interrupção enquanto esta sub-rotina está sendo executada, dizemos que houve uma Interrupção da Interrupção e este novo desvio somente ocorrerá se a Prioridade da segunda Interrupção for mais alta do que a que está sendo executada.

Se duas interrupções são solicitadas simultaneamente, a de prioridade mais alta é atendida primeiro. Se duas interrupções são de mesma prioridade e simultâneas, um sistema interno de prioridade é atendido .

- **Prioridade de Interrupção:**

Cada fonte de interrupção pode também ser individualmente programada para um de dois níveis de prioridade modificando-se um bit do SFR chamado Registrador IP (Interrupt Priority) .

Uma interrupção de baixa prioridade (bit 0) pode ser interrompida por uma de alta (bit 1), no entanto uma interrupção de alta prioridade não pode ser interrompida por qualquer outra fonte de interrupção.

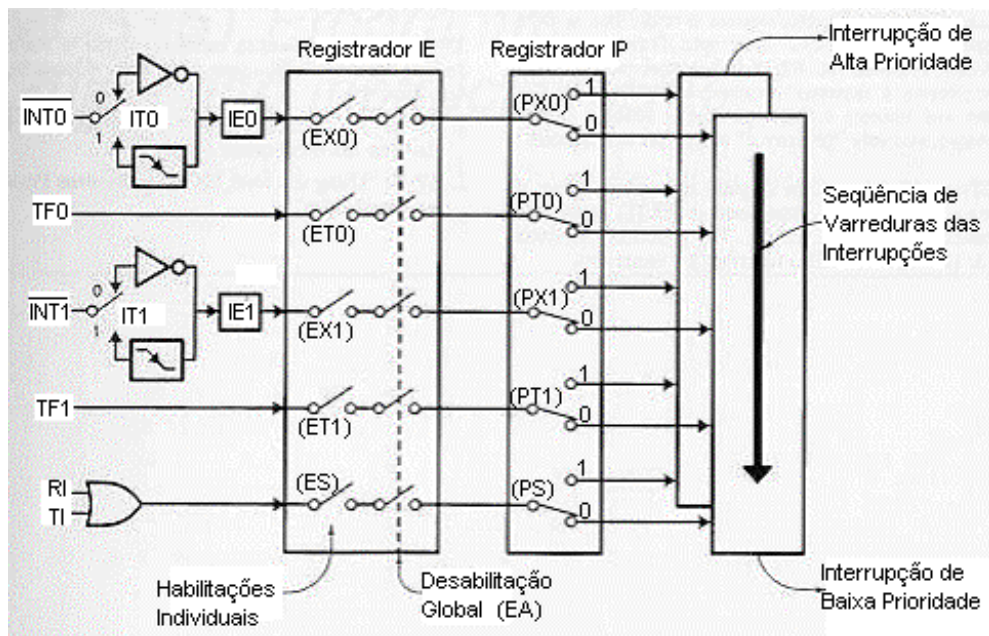
Registrador IP: (endereçável a Bit)

(MSB)								(LSB)
—	—	PT2	PS	PT1	PX1	PT0	PX0	

Bit = 1 --> Alta Prioridade
Bit = 0 --> Baixa Prioridade

Símbolo	Posição	Função
—	IP.7	
—	IP.6	
PT2	IP.5	Bit de Prioridade da Interrupção do Timer 2
PS	IP.4	Bit de Prioridade da Interrupção Serial
PT1	IP.3	Bit de Prioridade da Interrupção do Timer 1
PX1	IP.2	Bit de Prioridade da Interrupção Externa 1
PT0	IP.1	Bit de Prioridade da Interrupção do Timer 0
PX0	IP.0	Bit de Prioridade da Interrupção Externa 0

- **Sistema Interno de Prioridade de Interrupção:**



Exemplo : Se fizermos no Registrador IE:

EA = 1 → Habilita-se o uso das Interrupções

EX0 = 1 → Habilita-se a Interrupção Externa 0

ET0 = 1 → Habilita-se a Interrupção do Timer 0

e no Registrador IP:

Px0 = 0 → Baixa Prioridade para a Interrupção Externa 0

Pt0 = 0 → Baixa Prioridade para a Interrupção do Timer 0

E ocorrer uma solicitação simultânea da Interrupção Externa 0 e do Timer 0, a seqüência de atendimento será, primeiro a Interrupção Externa 0 depois a do Timer 0 .

- **Para usar as interrupções do MCS-51 , seguir os seguintes passos :**

1. Setar o bit EA (Enable All) do registrador IE.
2. Setar o bit do registrador IE correspondente à interrupção utilizada.
3. Escrever a sub-rotina de atendimento de interrupção no endereço correspondente.

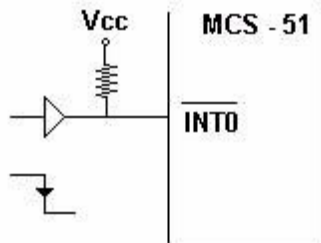
Interrupt Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI & TI	0023H
TF2 & EXF2*	002BH

4. estabelecer para as interrupções externas o tipo de disparo, nível ou descida de borda; para isso deve-se programar os bits IT0 e/ou IT1 do registrador TCON :

Registrador TCON: (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 sub-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 sub-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.					

Exemplo : Programação da Interrupção Externa 0 sensível à descida de borda.



Programa principal:

```
org 100h      ; Origem do Programa fora da área de Interrupções

setb EA      ; Habilita o uso de Interrupções
setb IE.0    ; Habilita a Interrupção Externa 0
setb IT0     ; Estabelece que deve ser sensível a descida de borda
...
...          ; Comandos do Programa Principal
...
end
```

Sub-rotina de Atendimento da Interrupção:

```
org 0003h    ; Endereço da Sub-rotina de Atendimento da Interrupção Externa 0.

clr EA       ; Desabilita as Interrupções para evitar Interrupção da Interrupção
push PSW    ; Salva os Flags do Programa Principal na pilha
...
...         ; Comandos da Sub-rotina de Atendimento da Interrupção
...
pop PSW     ; Recupera os Flags do Programa Principal
setb EA     ; Re-habilita as interrupções antes de voltar ao Programa Principal
reti       ; Volta para o Programa Principal
```

2.6 Temporizadores e Contadores (Timer/Counter) T/C

O 8051 possui 2 T/C internos, programáveis e com capacidade de operarem independentes da CPU . Podem ser habilitados ou desabilitados por Software ou por Hardware.

São 2 os registradores que comandam a programação dos T/C: o TCON e o TMOD

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 sub-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 sub-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.					

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

Dividido em duas partes com igual significado: uma para o Timer 0 e outra para o Timer 1.

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

GATE → Programável para se escolher o modo como os T/C deverão ser habilitados.

Para o Timer 1:

Controle por Hardware:

Quando o Bit TR1 (em TCON) é 1 e GATE=1, o T/C1 contará enquanto o pino INT1 na CPU for 1 .

Controle por Software:

Quando o GATE=0, o T/C1 contará enquanto o Bit TR1 (em TCON) = 1

Para o Timer 0:

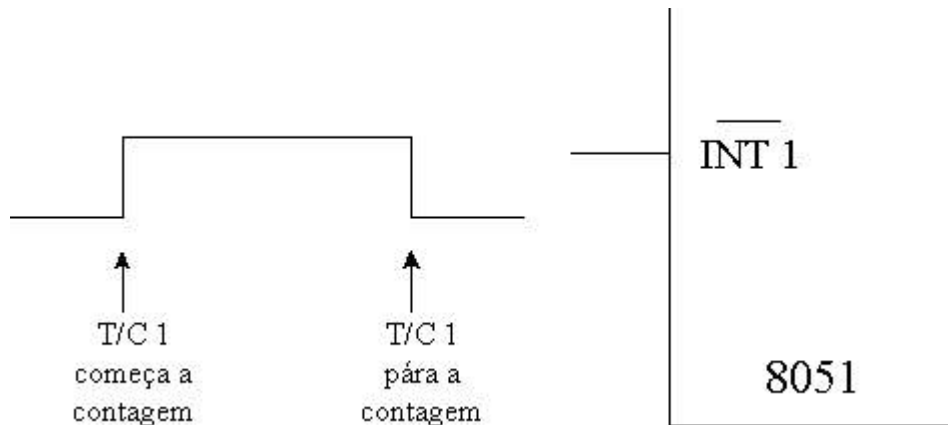
Controle por Hardware:

Quando o Bit TR0 (em TCON) é 1 e GATE=1, o T/C0 contará enquanto o pino INT0 na CPU for 1 .

Controle por Software:

Quando o GATE=0, o T/C0 contará enquanto o Bit TR0 (em TCON) = 1

Exemplo:
Medida de Largura de Pulso usando o T/C1: (Controle por Hardware)



No programa:

```
mov TMOD,#1xxxxxxb ; programa o T/C1 para controle por Hardware
setb TR1            ; ativa o T/C1
```

Obs: No exemplo, os outros bits de TMOD marcados com xxx não são irrelevantes mas estabelecem o tipo de contador a ser usado .

C/T → Bit Seletor de Temporizador (0=Timer) ou Contador (1=Counter)

Quando este bit é = 0 → O circuito será usado como Timer (temporizador), ou seja, a entrada de pulsos será através do clock interno ((frequência do oscilador)/12).

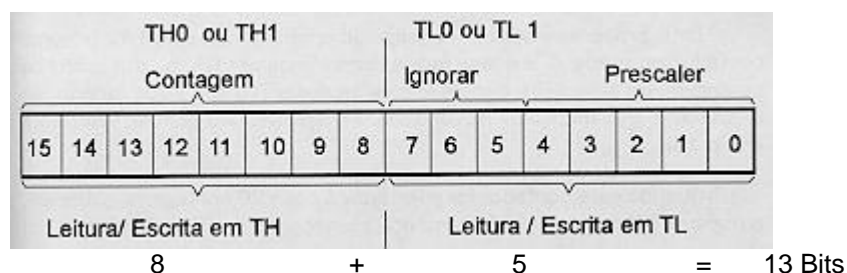
Quando este bit é = 1 → O circuito será usado como Counter (Contador), ou seja, a entrada de pulsos será pelos pinos T0 ou T1 da CPU.

M1 e M0 → Bits Seletores de Modo. Definem de quantos bits será o Contador/Temporizador.

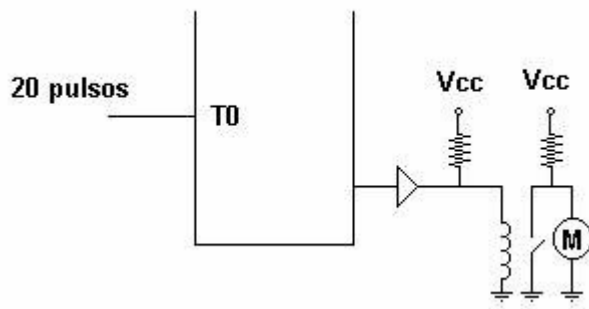
M1	M0	MODOS DE OPERAÇÃO	
0	0	0	Contador/Temporizador de 13 Bits (8 Bits + 5 Bits de Prescaler)
0	1	1	Contador/Temporizador de 16 Bits
1	0	2	Contador/Temporizador de 8 Bits Auto-recarregável
1	1	3	Para o TC0: TL0 é um Contador/Temporizador de 8 Bits controlado pelos bits de controle do TC0 e TH0 é um Contador/Temporizador de 8 Bits controlado pelos bits de controle do TC1
1	1	3	O TC1 não funciona no Modo 3

• **Exemplos de Operação em Cada Modo :**

1. Modo 0



Exemplo : Usando o T/C0 como contador no Modo 0, para contar de 20 em 20 pulsos 100 vezes e, após a contagem acionar um mecanismo que liga um motor ligado no bit 1 da porta P0..



Como o Contador é crescente (UP Counter) →

100 contagens -> 9BH___até___FFH (FFh – 9Bh = 64h → 100d)

```

mov TH0,#9Bh    ; contar cem vezes
mov TL0,#14h    ; Hexadecimal de 20d (contar vinte pulsos)
mov TMOD,#04    ; Timer 0 no modo 0 e como Contador
setb TR0        ; Inicializa o contador T/C0

```

Funcionamento :

O T/C0 conta em TL0 20 pulsos . A cada 20 pulsos é incrementado o valor de TH0 . Depois de 100 * 20 pulsos, ocorre um overflow em TH0 ocasionando TF0=1 . Para que o overflow funcione como solicitação de interrupção é necessário que anteriormente a interrupção equivalente esteja habilitada .

Ou seja, antes de disparar o contador, deve-se fazer :

```

setb EA ;habilita as interrupções
setb ET0 ;habilita Interrupção do Timer 0

```

Assim , quando TF0=1 , o programa salta para o endereço 000BH e poderá por exemplo acionar o bit na porta para controlar um mecanismo.

```

000BH: setb P1.0 ;Acione o mecanismo
        reti    ;Retorna

```

2. Modo 1

Nesse modo , o TH e o TL funcionam como um contador de 16 Bits (de 0000 a FFFFh) sendo que o valor inicial pode ser programado por Software. Quando o contador chegar em FFFFh, o próximo pulso gera um overflow e se a interrupção estiver habilitada, o programa salta para o endereço correspondente.

Exemplos:

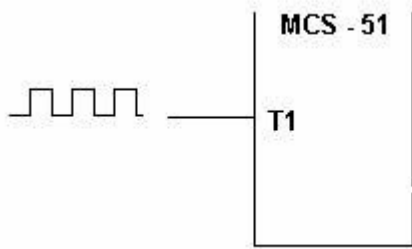
a. Programando o T/C1 como um temporizador (Timer) no Modo 1 com controle por Software .
(A frequência será 12 MHz/12 = 1 MHz)

```

mov TMOD,#00010000b ;10h
mov TH1,#80h        ;Contar de 8000h até FFFFh
mov TL1,#00         ;
setb TR1            ;Inicia a contagem

```

b. Programando o T/C1 como Contador (Counter) no Modo 1 com controle por Software:
A entrada de pulsos deverá ser aplicada em T1.



```

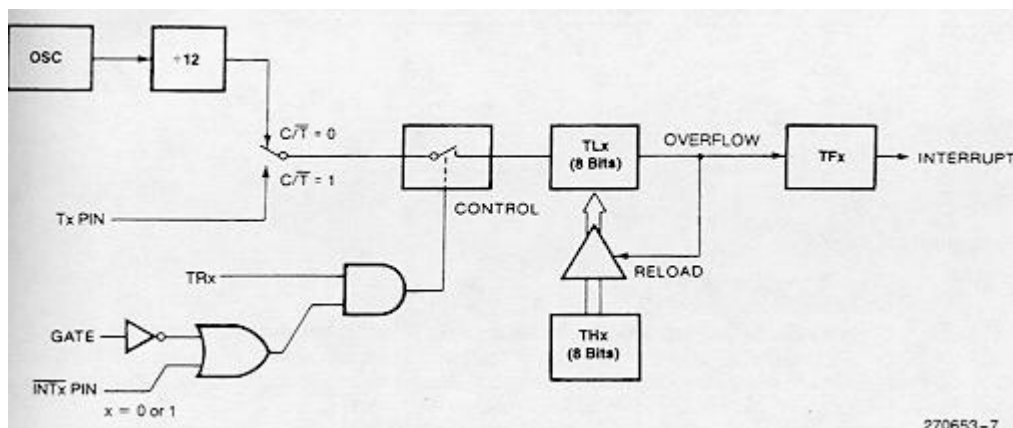
mov  TMOD,#01010000b ; 50h
setb TR1                ; Inicia a contagem
..
..
..

```

3. MODO 2

Contador de 8 Bits auto-recarregável.

Diagrama para o T/C1 no Modo 2:



Exemplos:

a. Temporizador de 8 Bits no Modo 2 (de 7Fh a FFh) para o T/C1. Controle por software.

```

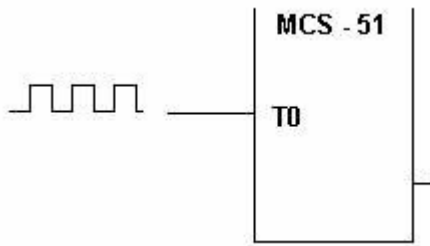
mov  TMOD,#00100000b ;20h
mov  TH1,#7Fh
mov  TL1,#7Fh
setb TR1                ;Inicia a contagem

```

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

Obs: Em qualquer dos modos, em qualquer instante pode-se ler o valor dos contadores, durante o tempo de contagem (em vôo), bastando para isso ler os valores de TH e TL .

b. Contador de 8 Bits no Modo 2 (de 00 a FFh) para T/C0.

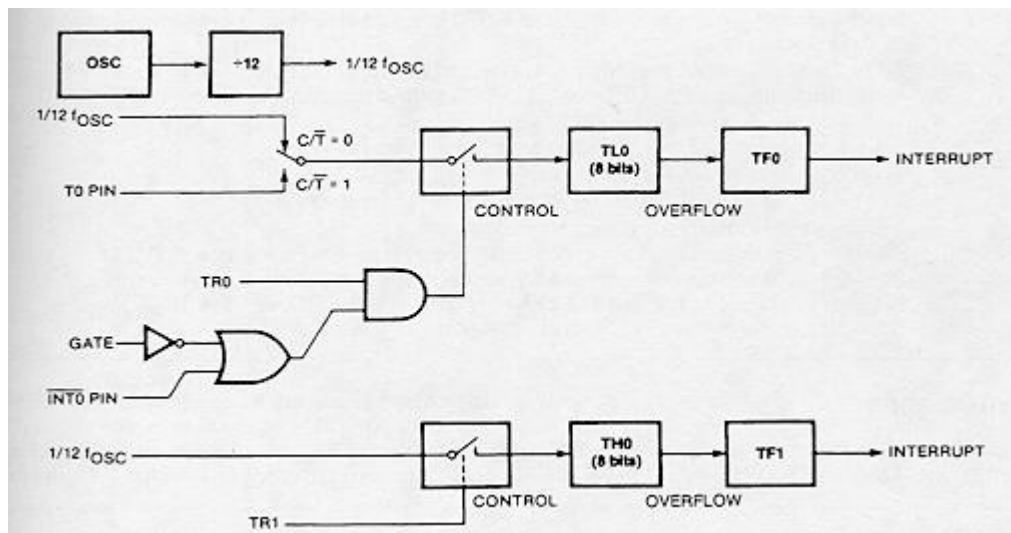


```

mov    TMOD,#01100000b    ;70h
mov    TH0,#00
mov    TL0,#00
setb   TR0                ; Inicia a Contagem
    
```

Modo 3: O T/C1 não funciona no Modo 3. O T/C0 no Modo 3 estabelece TL0 e TH0 como dois contadores separados de 8 Bits cada.

Diagrama para o T/C0 no Modo 3 :



Obs : Como o T/C1 não opera neste modo, TH0 gera uma solicitação de Interrupção através de TF1. O T/C1 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)

2.7 Interface Serial

A porta serial é full-duplex.

Escrever no SBUF (endereço do SFR = 99h), carrega o registrador de transmissão, e ler do SBUF (endereço do SFR = 99h) acessa um registrador de recepção fisicamente separado.

A porta serial pode operar de 4 Modos :

Modo 0 : O dado serial (de 8 Bits com LSB primeiro) é transmitido e recebido através de RxD. O TxD envia o clock. A Taxa de Comunicação (“baud rate”) é fixa em 1/12 da frequência do oscilador .

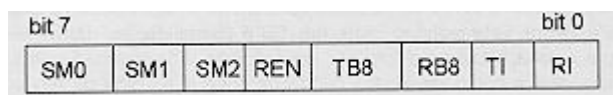
Modo 1 : 10 Bits são transmitidos (via TxD) ou recebidos (via RxD) : 1 StartBit, 8 Bits de dados com LSB primeiro e um StopBit. O “baud rate” é variável .

Modo 2 : 11 Bits são transmitidos (via TxD) ou recebidos (via RxD) : 1 StartBit, 8 Bits de dados com LSB primeiro, um nono Bit programável, e um StopBit . O “baud rate” é programável para 1/32 ou 1/64 da frequência do oscilador.

Modo 3 : 11 Bits são transmitidos (via TxD) ou recebidos (via RxD) : 1 StartBit, 8 Bits de dados com LSB primeiro, um nono Bit programável, e um StopBit. O “baud rate” é variável .

A transmissão é iniciada quando se escreve em SBUF .

O Registrador de Controle da Porta Serial (SCON) – Endereçável a Bit



SM0	SM1	Modo de Funcionamento	Taxa de Transmissão
0	0	0	Fclock/12
0	1	1	Variável
1	0	2	Fclock/32 ou Fclock/64
1	1	3	Variável

SM2 - Habilita a comunicação em multiprocessadores nos modos 2 e 3 . No modo 0 , SM2 deve ser = a zero .

REN - Habilita a recepção serial → 1 Começa a recepção
→ 0 Desabilita a recepção

TB8 - É o nono bit de dados que será transmitido nos modos 2 e 3. Setado ou zerado por Software.

RB8 - É o nono bit de dado que foi recebido nos modos 2 e 3. No Modo 1, se SM2=0 → RB8 é o StopBit recebido. No Modo 0 RB8 não é usado.

TI - Flag de Interrupção da transmissão. Setado por Hardware no fim do oitavo bit no Modo 0, ou no começo do StopBit nos outros Modos.

RI - Flag de Interrupção da Recepção. Setado por Hardware no fim do oitavo bit no Modo 0, ou na metade do tempo do StopBit, nos outros Modos.

(Baud Rates) Taxas de comunicação:

Modo 0 : Baud Rate fixo.

Baud Rate = Freqüência do oscilador / 12

Modo 1 : Baud Rate variável

Pode ser gerado tanto pelo Timer 1 ou Timer 2 (8052)

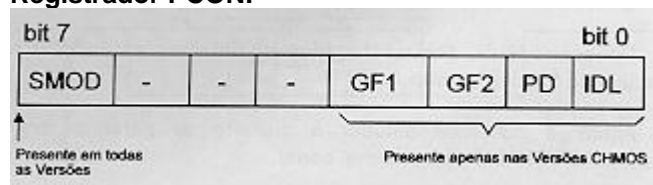
O T/C1 é usado no Modo 2 (Auto Reload) como Timer (C/T = 0)

$$BaudRate = \frac{K \times (Freqüência\ do\ Oscilador)}{32 \times 12 \times [256 - (TH1)]}$$

Se SMOD (de PCON) = 0 então K=1

Se SMOD (de PCON) = 1 então K=2

Registrador PCON:



TH1- (Byte mais significativo do Contador 1, que no Modo 2 é carregado em TL1 no fim de cada contagem)

TH1 é um valor inteiro (de 00 a 255)

O valor de TH1 pode ser calculado por:

$$TH1 = 256 - \frac{K \times (Freqüência\ do\ Oscilador)}{384 \times (Baud\ Rate)}$$

O arredondamento de TH1 para o inteiro mais próximo pode não produzir o Baud Rate desejado. Neste caso, deve-se escolher uma outra freqüência para o cristal .

Modo 2 :

Se SMOD = 1 -> Baud Rate = (freq. do oscilador) / 32

Se SMOD = 0 -> Baud Rate = (freq. do oscilador) / 64

Modo 3 :

Baud Rate igual ao do Modo 1

Por Exemplo:

Nos Modos 1 e 3, se quisermos gerar a Taxa de Comunicação de 19,2 KBPS (19.200 BPS) com uma freqüência do cristal de 12 MHz e SMOD = 1, podemos calcular o valor de TH1 e arredondá-lo para o inteiro mais próximo:

$$TH1 = 256 - \frac{2 \times 12 \times 10^6}{384 \times 19200} = 252,74 \cong 253 \Rightarrow 0FDh$$

Como TH1 deve ser inteiro, deve-se ajustar a frequência do cristal :

$$\text{Frequência do Oscilador} = \frac{\text{BaudRate} \times 384 \times (256 - TH1)}{K}$$

$$\text{Frequência do Oscilador} = \frac{19200 \times 384 \times (256 - 253)}{2} = 11059200 \text{ Hz} = 11.059 \text{ MHz}$$

Frequência do oscilador = 11.0592 MHz

Exercício : Qual a máxima Taxa de Comunicação em cada um dos modos, considerando o cristal de 12 MHz?

Resposta:

Taxa de Comunicação (BR = Baud Rate)

Modo 0 : BR = $fc/12 = 12/12 = 1\text{MBPS}$ (Mega Bits Por Segundo)

Modo 2 : BR = $fc/32 = 12/32 = 375\text{KBPS}$ (Kilo Bits Por Segundo)

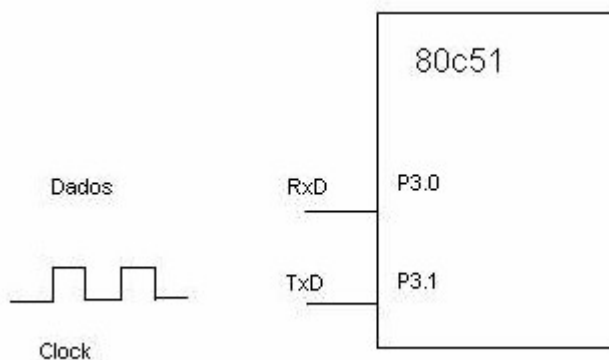
Modo 1 e 3 :

SMOD = 1 e TH1 = 0FFh (255)

$$BR = \frac{2 \times 12 \times 10^6}{32 \times 12 \times (256 - 255)} = 62,5 \text{ KBPS}$$

Detalhamento da Operação em cada Modo :

Modo 0 : Comunicação Serial Síncrona



Os dados são transmitidos e recebidos através de RxD (P3.0). O pino TxD (P3.1) transmite o clock .

Para transmitir um dado:

1. A transmissão inicia ao se escrever o dado em SBUF

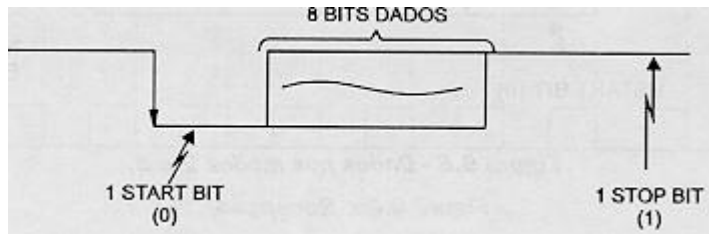
```
mov SBUF,#dado
```

2. O dado é transmitido serialmente na frequência de 1/12 da frequência do oscilador.
3. Quando o MSB do dado for transmitido o bit TI é setado .

Para Receber um dado:

1. A Recepção começa se REN = 1 e RI = 0 .
2. Quando os 8 Bits do dado forem recebidos, eles são transferidos para SBUF.
3. O bit RI é setado .

Modo1: Comunicação serial Assíncrona. (10 Bits)



Na recepção o StopBit vai para RB8 do SCON

Transmissão :

1. A transmissão começa escrevendo-se o dado em SBUF

```
mov SBUF,#dado
```

2. Quando o MSB foi transmitido o BIT TI de SCON é setado.

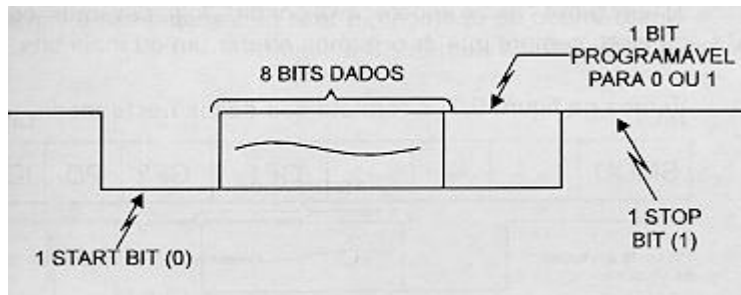
Recepção :

1. Iniciada quando uma transição de 1 para 0 é detectada em RxD . A linha é amostrada 16 vezes. O valor é aceito após 2 ou 3 amostras para evitar ruído (falso StartBit) .
2. Os dados recebidos são destacados por um Shift Register e ao final carregado em SBUF, o StopBit em RB8 e RI é setado .
3. Isto só ocorre se a seguinte condição existir no final da recepção

- a. Ri =0
- b. SM2 ou StopBit recebido = 1

Se estas condições não ocorrerem o dado foi perdido .

Modo 2 e Modo 3 : Comunicação Serial Assíncrona (11 Bits)



Na Transmissão: ao nono Bit (TB8) pode ser atribuído o valor 0 ou 1 .

Na Recepção: o nono Bit vai para RB8 em SCON .

Transmissão:

1. A Transmissão começa quando se escreve em SBUF o dado a ser transmitido. Esta escrita automaticamente carrega TB8 no nono Bit .

```
mov    SBUF,#dado
```

2. Quando o TB8 é transmitido pelo nono Bit, TI é setado .

Recepção:

1. A Recepção é iniciada quando uma transição de 1 para 0 é detectada em RxD. Da mesma forma que no Modo 1, a linha é amostrada 16 vezes para evitar StartBit falso.

2. Quando todo o carácter é deslocado, o dado é carregado em SBUF e o nono Bit em RB8, setando RI .

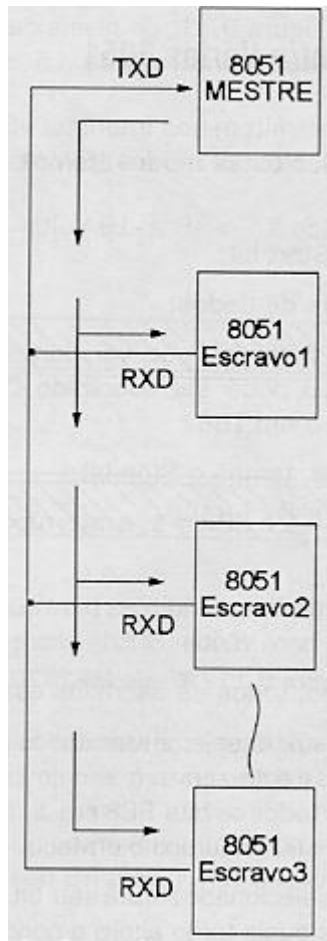
3. Isto só acontece se :

a. RI =0

b. Tanto SM2 = 0 ou o nono Bit recebido = 1

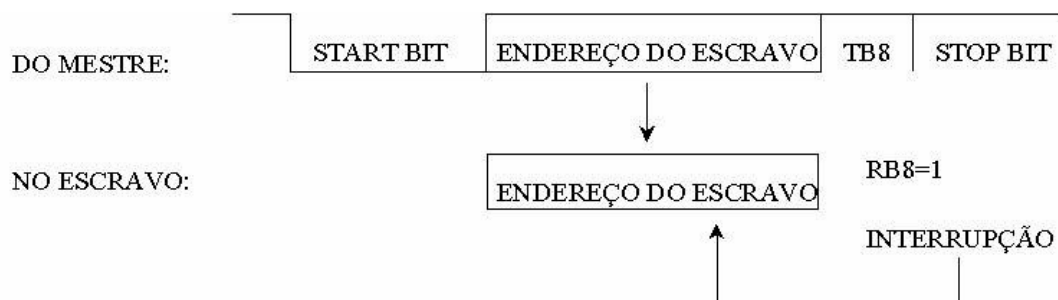
Comunicação com Microprocessadores

Usando os Modos 2 e 3 , é possível interligar vários 8051 em rede (1 mestre e vários escravos)



Quando $SM2 = 1$ e $RB8 = 1$, a interrupção do canal serial será atendida.

1. No início, todos os escravos devem Ter $SM2 = 1$
2. Quando o mestre desejar enviar dados para um dos escravos faz $TB8 = 1$ e envia o endereço do escravo com o qual quer se comunicar. Dessa forma o $RB8$ de todos os escravos será igual a 1, solicitando interrupção em cada um .



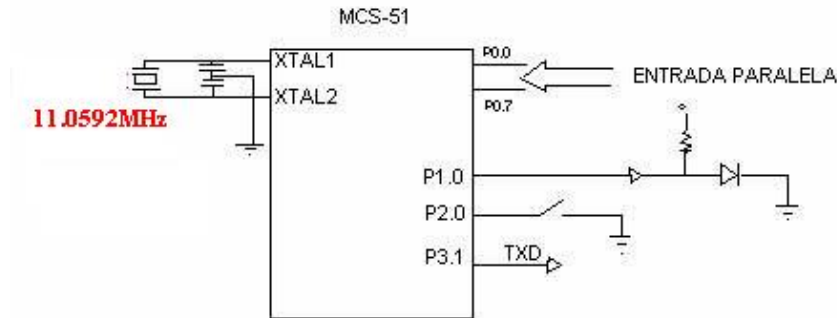
3. A sub-rotina de atendimento de interrupção verifica o endereço enviado. O escravo selecionado faz $SM2 = 0$ ficando preparado para receber os dados .
4. Cada dado enviado pelo mestre para o escravo selecionado terá $RB8 = 0$.
5. Os demais escravos continuarão com $SM2 = 1$ e não serão mais interrompidos pois os dados tem $RB8 = 0$.

Ou seja, endereço do escravo é enviado com o nono Bit setado e dados são enviados com o nono bit zerado .

Quando o mestre quiser enviar dados para outros escravos basta enviar o endereço deste escravo com o nono bit em 1 e recomeçar o processo .

Exemplo de Transmissão Serial (1200 BPS) Assíncrona

Escrever um programa que transmita pela saída Serial, qualquer dado (8 bits em paralelo) que entre pela Porta P0, dependendo de uma chave de autorização de transmissão ligada em P2.0. Assim que o Dado for transmitido, avisar através de um Led conectado na porta P1.0.



CAPACITORES = 33 pF

```
org 0
sjmp PROG
```

```

;*****
;
; sub-rotina de atendimento da Interrupção Serial
;*****
;
    org 0023h    ;endereço da interrupção serial
                ;salta para aqui quando acabar a transmissão de
                ;um byte

    clr  TI     ;zerar o flag de transmissão
    setb P1.0  ;avisa que o byte já foi transmitido
    reti

;*****
;
; Programa Principal
;*****
PROG:    mov  IE,#90h    ;IE = 10010000 → habilitação da interrupção serial
        mov  TMOD,#20h ;TMOD = 00100000 →Timer 1 no Modo 2, controle por software
        mov  TH1,#0E8h ;valor E8h em TH1 para gerar a Taxa de Comunicação de
        mov  TL1,#0E8h ;1200 BPS com fc=11.059 MHz
        setb TR1       ;Dispara Timer
        mov  SCON,#40h ;SCON = 01000000 → modo 1 do Canal Serial
aqui:   jb  P2.0,aqui   ;Verifica se pode transmitir
        mov  SBUF,P0   ;Realiza a transmissão
        sjmp aqui     ;Continua a transmitir o dado que entra pela Porta P0
;*****
end

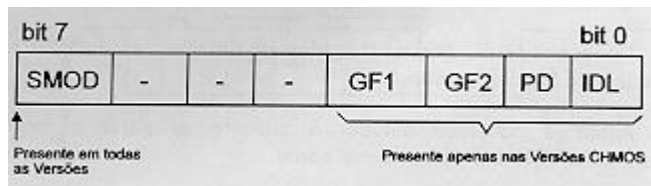
```

2.8) Controle de Potência no 8051

As versões CHMOS possuem dois modos de controle de redução de potência .

1. Modo Idle
2. Modo Power Down

O registrador de controle é o PCON



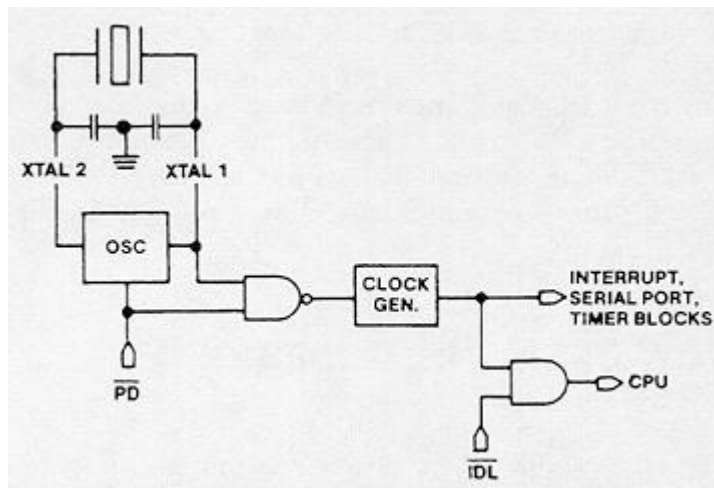
SMOD -- comunicação serial

GF1 e GF0 -- bits de uso geral

PD -- Bit de Power Down

IDL -- Bit de Modo Idle

PD tem precedência sobre IDL



No Modo Idle, (IDL = 1) o oscilador continua a rodar e as interrupções, a Porta Serial e os Timers operam normalmente, mas o sinal de clock para a CPU pára.

No Modo Power Down (PD = 1) o oscilador pára.

Nos dispositivos HMOS o registrador PCON contém apenas o SMOD .

Modo Idle

Escrever 1 em IDL faz com que a CPU entre no Modo Idle. O status da CPU é preservado : o SP, o PC, o PSW, o ACC e todos os outros registradores mantêm seus dados durante o tempo em que a CPU estiver em Idle.

As Portas mantêm seus estados lógicos que tinham quando o Modo Idle foi ativado .

Existem 2 Modos de terminar o Modo Idle

1. Ativação de uma Interrupção habilitada
2. Reset de Hardware

Modo Power Down

Escrever 1 em PD faz com que a CPU vá para o Modo Power Down. Neste Modo o oscilador pára. Todas as funções param. O conteúdo da RAM interna e dos SFRs se mantém. A única forma de sair deste modo e através do reset de hardware . Neste Modo Vcc pode ser reduzido para 2 Volts.

Dispositivos CHMOS (CMOS)

1. No 80C51 drenam menos corrente que os HMOS
2. No modo "repouso" (Idle mode) a CPU é desligada, mas a RAM e os chips periféricos continuam operando. Neste modo o consumo é reduzido em 85%
3. No modo "desligado" (Power down mode) todas as atividades são suspensas, só os dados da RAM são mantidos. O consumo é da ordem de 10 μ A