CAPÍTULO VIII

TEMPORIZADORES / CONTADORES

8.1. INTRODUÇÃO

O 8051 tem dois registros contadores de 16 bits, denominados TIMER 0 e TIMER 1, dedicados às funções de contagem e temporização (counter/timer-contador/temporizador). Há uma importante distinção entre os conceitos de contador e temporizador.

Quando opera como **temporizador**, o registro é incrementado a cada ciclo de máquina (usa como base o cristal da CPU). O sinal de contagem aparece com 1/12 da freqüência do clock.

Quando opera como **contador**, o registro é incrementado de acordo com o sinal que se coloca nas entradas T1 e T0, ou seja, o contador opera a cada transição de 1 para 0 (borda de descida ↓) na entrada T0 ou T1.

Deve-se ter um cuidado quando em operações em modo contador: já se sabe que as entradas são amostradas durante S5P2 de cada ciclo de máquina (figura 3.5). Quando em um ciclo de máquina a entrada é detectada em 1 e depois em 0, no próximo ciclo o contador é incrementado. O incremento acontece em S3P1 do ciclo seguinte ao que foi detectada a transição. Para garantir que o nível correto tenha sido amostrado, é necessário que o sinal de entrada (o sinal que vai acionar os contadores) permaneça pelo menos um ciclo de máquina em nível alto e pelo menos outro ciclo de máquina em nível baixo, quer dizer, a máxima freqüência que responderá o contador é de 1/24 da freqüência de clock. O sinal de entrada pode ter qualquer "duty cycle", desde que se respeite a restrição antes mencionada.

8.2. REGISTROS ENVOLVIDOS

Dois registros SFR são utilizados para controlar as funções e operações do temporizador/contador: em **TMOD** especifica-se o modo de operação e em **TCON** controla-se a operação. A figura 8.1 ilustra o registro **TMOD** e a figura 8.2 ilustra o registro **TCON**.

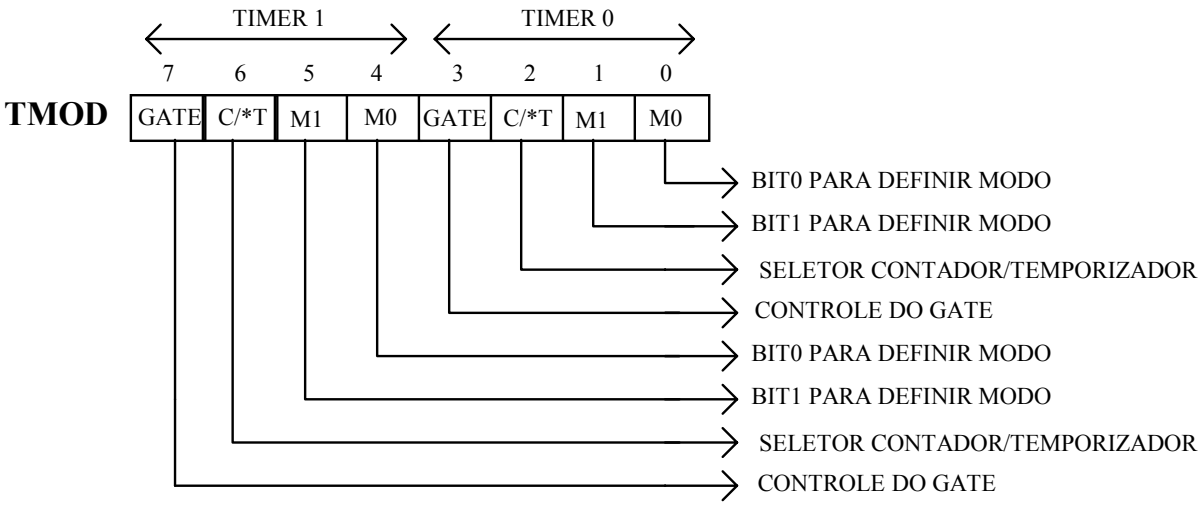


Figura 8.1. Descrição do registro TMOD.

GATE → especifica como será feito o controle:

se GATE = 1 → conta somente se TR1=1 e INT1=high, (idem para TR0 e INT0)

se GATE = 0 → conta somente se TR1=1 (controle somente por software)

(GATE especifica se INT1 será usado para controlar o funcionamento do contador/temporizador ➔ pode ser usado para medir a largura de pulsos externos ligados a INT0 ou INT1)

C/*T → seleciona modo contador ou temporizador:

se C/*T = 1 → modo contador (conta usando a entrada T1)

se C/*T = 0 → modo temporizador (conta a cada ciclo de máquina)

M1 M0 → seleciona o modo de operação:

0 0 → THi é temporizador/contador de 8 bits e TLi é um pre-scaler de 5 bits,

0 1 → THi e TLi formam um temporizador/contador de 16 bits,

1 0 → contador/temporizador de 8 bits com auto-recarga (TLi conta e THi valor para recarga),

1 1 → TL0 contador/temporizador de 8 bits (usando TR0, *INT0 e TF0)

→ TH0 contador/temporizador de 8 bits (usando TR1, *INT1 e TF1)

→ TH1 e TL1 parado (mas pode operar em outros modos)

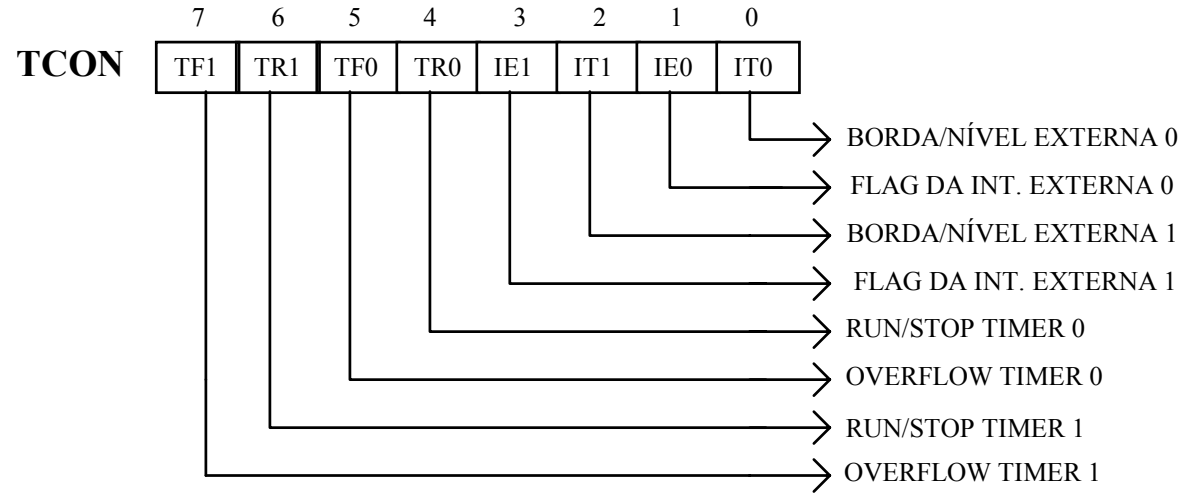


Figura 8.2. Descrição do registro TCON.

TF1 → flag de transbordamento (overflow) do contador/temporizador 1. Ativado por hardware quando há transbordamento no timer 1. É apagado por hardware quando o processamento é desviado para a rotina de interrupção.

TR1 → bit de partida/parada (run/stop) do contador/temporizador 1.

TF0 → flag de transbordamento (overflow) do contador/temporizador 0. Ativado por hardware quando há transbordamento no timer 0. É apagado por hardware quando o processamento é desviado para a rotina de interrupção.

TR0 → bit de partida/parada (run/stop) do contador/temporizador 0.

IE1 → flag da interrupção externa 1. Ativada por hardware quando é detectada uma interrupção. Apagado por hardware (somente se for modo borda) quando a interrupção é processada, ou seja, quando se desvia para a rotina de interrupção.

IT1 → indica se a interrupção externa 1 opera por borda ou por nível:

1 → borda de descida (↓),

0 → nível baixo.

E0 → flag da interrupção externa 0. Ativada por hardware quando é detectada uma interrupção. Apagado por hardware (somente se for modo borda) quando a interrupção é processada, ou seja, quando se desvia para a rotina de interrupção.

IT0 → indica se a interrupção externa 0 opera por borda ou por nível:

1 → borda de descida (↓),

0 → nível baixo.

8.3. Modos de operação

Os dois contadores/temporizadores (timer1 e timer0) podem trabalhar em 4 modos de operação que são selecionados empregando os bits M1 e M0 do registro TMOD. Os modos 0, 1 e 2 são iguais para os 2 contadores/temporizadores mas o modo 3 é diferente.

8.3.1. Modo 0

Este modo é idêntico para os dois contadores/temporizadores. Neste modo tem-se um contador de 8 bits com um divisor (pre-scaler) de 5 bits. Resulta então em um contador/temporizador de 13 bits, compatível com o que havia no MCS-48. Os 13 bits são formados pelos 8 bits do registro THi e pelos 5 bits menos significativos do registro TLi. Os outros 3 bits do TLi são indeterminados. O transbordamento (overflow) é gerador quando a contagem faz a transição de 1FFFH para 0000; neste instante ativa-se o bit de overflow (TF1 ou TF0).

O controle da contagem é simples:

- se GATE = 0, TR1 controla o contador/temporizador (controle por software)
- se GATE = 1, TR1 e INT1 controlam o contador/temporizador
  (permite também um controle externo por hardware)

A figura 8.3 apresenta um diagrama em blocos do contador/temporizador 1 operando em modo 0. A mesma figura é válida para o contador/temporizador 0.

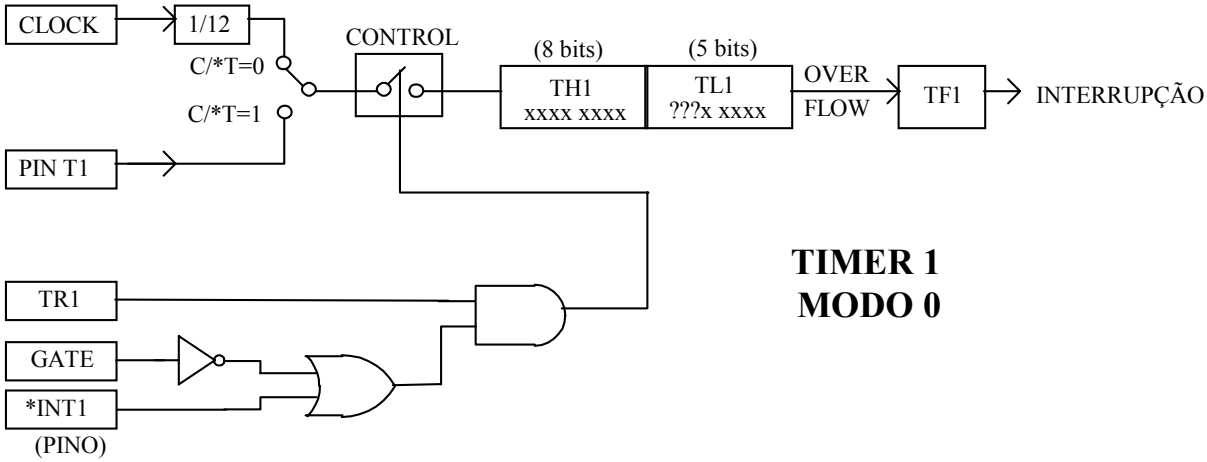


Figura 8.3. Diagrama em blocos para o TIMER 1 em MODO 0.

8.3.2. Modo 1

Este modo de operação é o mais simples e por isto é muito utilizado. É idêntico ao modo 0, mas os contadores são de 16 bits. A figura 8.4. ilustra o diagrama em blocos para este modo, que é idêntico para os dois timers.

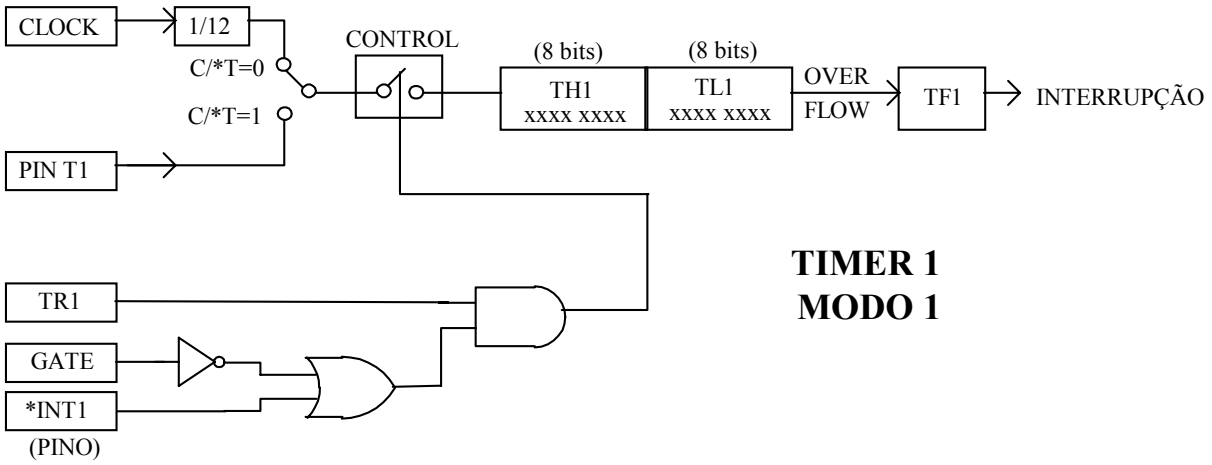


Figura 8.4. Diagrama em blocos para o TIMER 1 em MODO 1.

8.3.3. Modo 2

Neste modo tem-se um contador/temporizador de 8 bits (TLi) com registro de recarga (THi) para quando ocorre o transbordamento. Neste modo os dois contadores/temporizadores operam de forma idêntica. O transbordamento (overflow) não somente ativa TFi como também recarrega TLi com o valor guardado em THi (este permanece inalterado). O valor de recarga deve ser fornecido por software.

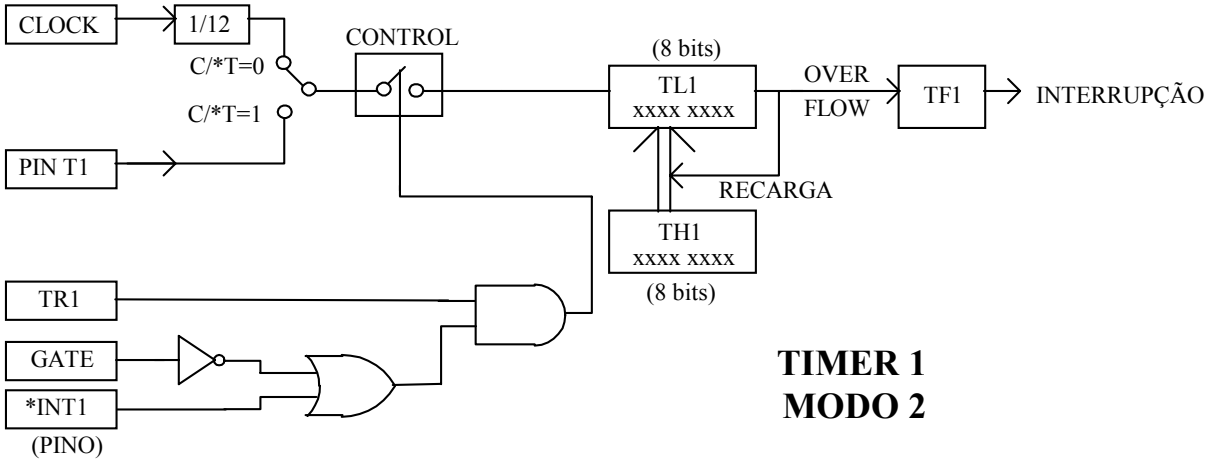


Figura 8.5. Diagrama em blocos para o TIMER 1 em MODO 2.

8.3.4. Modo 3

Este é o único modo onde os contadores/temporizadores têm um comportamento diferente. Neste modo, o contador/temporizador 1 simplesmente suspende a contagem (é como se estivesse com TR1=0), enquanto o contador/temporizador 0 se divide em dois contadores de 8 bits:

TH0 → contador/temporizador de 8 bits usando C/*T, GATE, TR0, *INT0 e TF0,

TL0 → contador/temporizador de 8 bits usando TR1 e TF1 (ou seja, provoca a interrupção do timer 1).

É como se existem 2 contadores/temporizadores de 8 bits (TH0 e TL0) e outro de 16 bits (timer 1). O timer 1, entretanto, pode ser usado para operar em qualquer outro modo (0, 1 ou 2) e usa-se o modo 3 para detê-lo. Não há problemas porque os modos são independentes. Também se pode usar o timer 1 (em modo 2) para gerar o "baud rate" da porta serial.

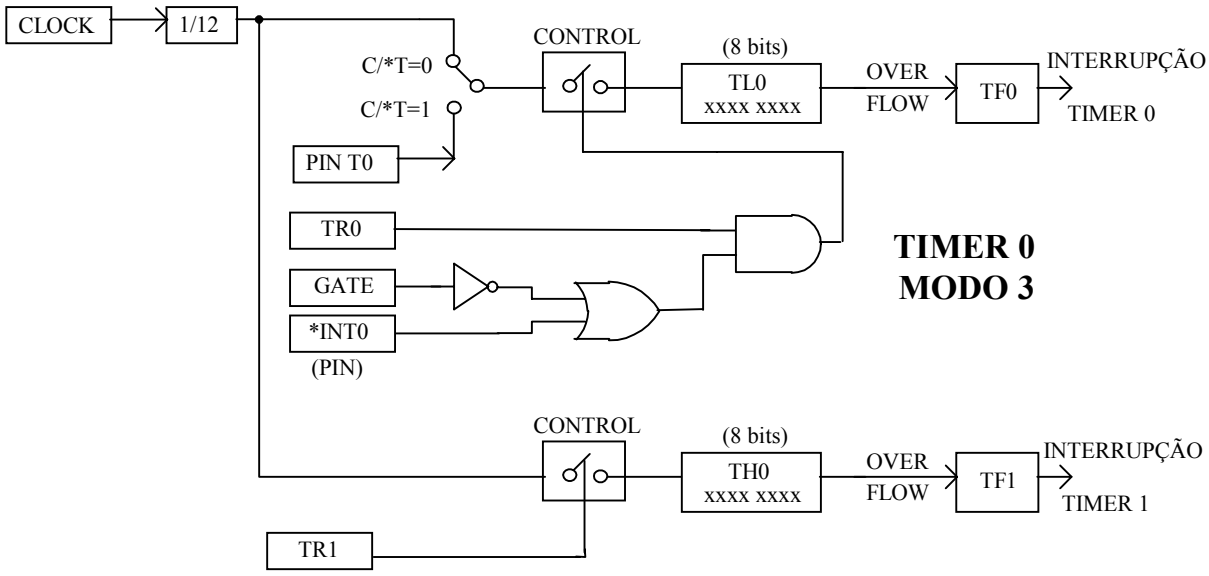
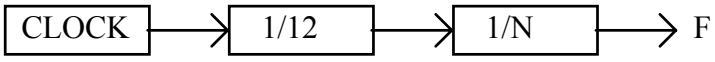


Figura 8.6. Diagrama em blocos para o TIMER 0 em MODO 3.

8.4. EXERCÍCIOS

EXERCÍCIO 8.1. LED_10HZ

Acender os leds vermelho, amarelo e verde a uma freqüência de 10 Hz. Para solucionar esse exercício é necessário gerar um retardo equivalente ao período de 10 Hz. A figura 8.7 ilustra o cálculo do valor do contador para gerar esse retardo.



$$F = \frac{CLOCK}{12*N} \qquad N = \frac{CLOCK}{12*F}$$

Figura 8.7. Cálculo de N para gerar uma determinada freqüência F.

Se F=10 Hz e CLOCK=3,575611 Hz, então N=29797. Como este é um contador que conta para cima, temos que subtrair este valor de 65536. Com isto, N=65536-29797=35739 ou 8B9BH. Na realidade, a freqüência gerada não será exatamente 10 Hz mas de um valor muito próximo:

F = 3,575611 / 12*29797 = 9,99 Hz

Será utilizado o timer 0 no modo 1; a cada interrupção a rotina acende um novo led e recarrega o timer.

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

Figura 8.8. Valor a ser programado em TMOD.

IE	EA	-	-	ES	ET1	EX1	ET0	EX0
	1	0	0	0	0	0	1	0

Figura 8.9. Valor a ser programado em IE.

```
;LED_10HZ.ASM
;
                DEFSEG PROG, CLASS=CODE, START=0
                SEG             PROG
;
DEZ_HZ          EQU             35739
;
                ORG             RESET
                AJMP            INIC
;
                ORG             TIMER0
                AJMP            TIM0
;
                ORG             50H
INIC            MOV             TL0,#LOW  DEZ_HZ
                MOV             TH0,#HIGH DEZ_HZ
                MOV             TMOD,#1         ;TIMER 0 EM MODO 1
                MOV             IE,#82H
                MOV             P1,#0           ;APAGAR TODOS OS LEDS
                MOV             A,#01001001B
                CLR             C
                SETB            TR0
                SJMP            $
;
                ORG             100H
TIM0            MOV             TL0,#LOW  DEZ_HZ  ;REINICALIZAR
                MOV             TH0,#HIGH DEZ_HZ  ;O CONTADOR
                RLC             A
                MOV             P1,A
                RETI
                END
```

Notar que na rotina de interrupção está a recarga dos temporizadores que é feita com a instrução "MOV direto,#data" mas que consome 2 ciclos de máquina, ou seja, que há um retardo que se acumula. Para evitar ou corrigir isso, pode-se adicionar ao contador o retardo da instrução; é mais preciso usar "MOV TL0,# LOW (DEZ_HZ+2).

Quando existem várias interrupções, não se sabe com certeza quando uma interrupção do temporizador pode ser aceita; isso implica um erro maior que se acumula. Há uma solução simples porque depois do transbordamento o temporizador segue contando e esta contagem é exatamente o tempo que atrasou o serviço da interrupção; basta usar esse valor na recarga. As instruções a seguir ilustram a idéia.

```
        MOV     A,#LOW (DEZ_HZ+2)
        ADD     A,TL0                   ;1 ciclo
        MOV     TL0,A                   ;1 ciclo
        MOV     TH0,#HIGH (DEZ_HZ+2)
```

EXERCÍCIO 8.2. LED_1HZ

Acender os leds vermelho, amarelo e verde a uma freqüência de 1 Hz. A solução é idêntica ao exercício anterior, mas vai surgir uma dificuldade ao calcular os valores de recarga:

CLOCK=3575611 Hz e F=1 Hz

N=3575611 / 12*1 = 297968

Como se pode ver, o valor de recarga é muito grande (297 968 > 65 536) e impede a utilização do temporizador. Uma boa saída é utilizar o esquema do exercício anterior (com 10 Hz) e adicionar, por software, um divisor por 10.

```
;LED_1HZ.ASM
;
                DEFSEG PROG, CLASS=CODE, START=0
                SEG             PROG
;
DEZ_HZ          EQU             32203
DIVISOR         EQU             10
;
                ORG             RESET
                AJMP            INIC
;
                ORG             TIMER0
                AJMP            TIM0
;
                ORG             50H
INIC            MOV             TL0,#LOW  DEZ_HZ
                MOV             TH0,#HIGH DEZ_HZ
                MOV             R7,#DIVISOR     ;PREPARAR DIVISAO POR 10
                MOV             TMOD,#1         ;TIMER 0 EM MODO 1
                MOV             IE,#82H
                MOV             P1,#0           ;APAGAR TODOS OS LEDS
                MOV             A,#01001001B
                CLR             C
                SETB            TR0
                SJMP            $
;
                ORG             100H
TIM0            MOV             TL0,#LOW  DEZ_HZ  ;REINICALIZAR
                MOV             TH0,#HIGH DEZ_HZ  ;O CONTADOR
                DJNZ            R7,FIM            ;DIVIDIR POR 10
                MOV             R7,#DIVISOR
                RLC             A
                MOV             P1,A
FIM             RETI
                END
```

EXERCÍCIO 8.3. ONDA1

Gerar através de P1.7 uma onda com o seguinte formato:

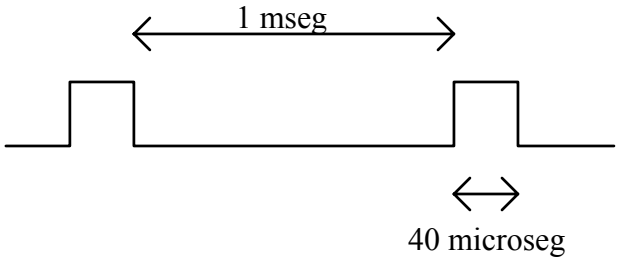


Figura 8.10. Sinal a ser gerado pela saída P1.7.

Usando N = (t*CLOCK)/12, calcula-se o valor de recarga para o temporizador. Serão programados dois retardos alternadamente de forma a gerar o sinal pedido.

Para t=1 ms $\rightarrow$ N=298 $\rightarrow$ 65536-298=65238

Para t=40 μs $\rightarrow$ N=12 $\rightarrow$ 65536-12 =65524

```
;ONDA1.ASM
                DEFSEG PROG, CLASS=CODE, START=0
                SEG PROG
;
SAIDA           EQU             P1.7
R_1MS           EQU             65238           ;RETARDO DE 1 MILISEG
R_40MICRO       EQU             65524           ;RETARDO DE 40 MICROSEG
;
                ORG             RESET
                AJMP            INICIO
;
                ORG             TIMER0
                AJMP            TIM0
;
                ORG             50H
INICIO          MOV             TL0,#LOW  R_1MS
                MOV             TH0,#HIGH R_1MS
                MOV             TMOD,#1         ;TIMER 0 EM MODO 1
                MOV             IE,#82H         ;EA=1 E ET0=1
                CLR             SAIDA
                SETB            TR0             ;PARTIDA DO TIMER 0
                SJMP            $               ;LOOP INFINITO
;
TIM0            JB              SAIDA,LB1                       ;2 CICLOS
                MOV             TH0,#LOW  (R_40MICRO+4)         ;2 CICLOS
                MOV             TL0,#HIGH (R_40MICRO+4)         ;2 CICLOS
                SETB            SAIDA                           ;1 CICLO
                RETI                                            ;2 CICLOS
LB1             MOV             TH0,#LOW  (R_1MS+4)
                MOV             TL0,#HIGH (R_1MS+4)
                CLR             SAIDA
                RETI
                END
```

Notar que a rotina que atualiza o temporizador para gerar o retardo de 40 μs consome 9 ciclos de máquina, quer dizer, consome 30,2 μs. Pode parecer que se o tempo de 40 μs fosse mudado para 25 μs não haveria solução.

EXERCÍCIO 8.4. ONDA2

Gerar através de P1.7 uma onda com o seguinte formato:

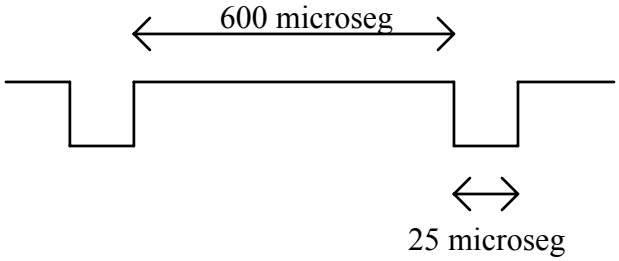


Figura 8.11. Sinal a ser gerado pela saída P1.7.

Usando N = (t*CLOCK)/12, calcula-se o valor de recarga para o temporizador. O segredo será trabalhar com o temporizador em modo 2 (os dois valores de recarga são menores que 256) e a rotina de interrupção apenas muda o valor da recarga (que está em TH0).

Para t=600 μseg → N=179 → 256-179=77

Para t=25 μseg → N=7 → 256-7 =249

```
;ONDA2.ASM
                DEFSEG PROG, CLASS=CODE, START=0
                SEG PROG
;
SAIDA           EQU             P1.7
R_600MICRO      EQU             77              ;RETARDO DE 600 MICROSEG
R_25MICRO       EQU             249             ;RETARDO DE 25 MICROSEG
;
                ORG             RESET
                AJMP            INICIO
;
                ORG             TIMER0
                AJMP            TIM0
;
                ORG             50H
INICIO          MOV             TL0,#R_600MICRO ;PRIMEIRA CONTAGEM
                MOV             TH0,#R_25MICRO  ;CONTAGEM SEGUINTE
                MOV             TMOD,#2         ;TIMER 0 EM MODO 2
                MOV             IE,#82H         ;EA=1 E ET0=1
                CLR             SAIDA
                SETB            TR0             ;PARTIDA DO TIMER 0
                SJMP            $               ;LOOP INFINITO
;
TIM0            JB              SAIDA,LB1       ;2 CICLOS
                MOV             TH0,#R_25MICRO  ;2 CICLOS
                SETB            SAIDA           ;1 CICLOS
                RETI                            ;2 CICLOS
LB1             MOV             TH0,#R_600MICRO
                CLR             SAIDA
                RETI
                END
```

EXERCÍCIO 8.5. LEDS1

Acender os leds vermelho, amarelo e verde em seqüência, mudando a cada 10 pulsos em T1. Será usado o contador/temporizador 1 como contador e programado para operar em modo 2, com um valor de recarga igual a 246 (256-10). O acumulador e carry serão usados para acender os leds na seqüência correta

TMOD	GATE	C/*T	M1	M0	GATE	C/*T	M1	M0
	0	1	1	0	0	0	0	01

Figura 8.12. Valor a ser programado em TMOD (contador 1, modo 2).

IE	EA	-	-	ES	ET1	EX1	ET0	EX0
	1	0	0	0	1	0	0	0

Figura 8.9. Valor a ser programado em IE.

```
;LEDS1.ASM
                DEFSEG PROG, CLASS=CODE, START=0
                SEG PROG
;
CONTA_10        EQU             246
;
                ORG             RESET
                AJMP            INICIO
;
                ORG             TIMER1
                AJMP            TIM1
;
                ORG             50H
INICIO          MOV             A,#01001001B
                CLR             C
                MOV             P1,A
                MOV             TL1,#CONTA_10   ;CARREGAR CONTADOR
                MOV             TH1,#CONTA_10   ;VALOR DE RECARGA
                MOV             TMOD,#60H       ;CONTADOR 1, MODO 2
                MOV             IE,#88H         ;EA=1, ET1=1
                SETB            TR1             ;LIGAR CONTADOR
                SJMP            $               ;LOOP INFINITO
;
TIM1            RLC             A
                MOV             P1,A
                RETI
                END
```

A chave SW3 aciona diretamente a entrada T1, mas há bouncing e por isso serão notadas mudanças nos leds antes de 10 acionamentos. Para esse caso específico o bouncing deverá ser eliminado por hardware.