

EXPERIÊNCIA 9

Contadores Assíncronos

1 Objetivos

- Estudar os contadores assíncronos.
- Implementar um contador assíncrono de 0 a 9.

2 Motivações

- Como obter um contador a partir da associação de diversos flip-flops?
- O que significa contador “assíncrono”?
- Há algum CI que já implementa um contador?
- Quais suas aplicações?

3 Referências para Consulta

http://pdf.datasheetcatalog.com/datasheet/on_semiconductor2/74HC74.PDF

pdf.datasheetcatalog.com/datasheet/philips/74HC_HCT93_CNV_2.pdf

pdf.datasheetcatalog.com/datasheets/50/231825_DS.pdf (7490)

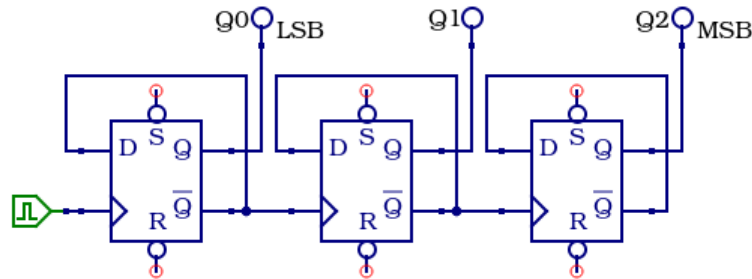


Figura 1: Contador assíncrono de 0 a 7 (em binário, de 000_2 a 111_2). (Nota: considere-se o \bar{S} e o \bar{R} conectados em nível lógico “1”.)

4 Contadores Assíncronos

4.1 Contextualização

Os contadores assíncronos são construídos pela ligação em “cascata” de vários *flip-flops* do tipo JK, D ou T. Ligar em “cascata” significa que o *clock* do *flip-flop* seguinte é obtido pela saída do *flip-flop* anterior. O desenho da figura 1 ilustra um exemplo.

Note-se que o gerador de sinal de *clock* é conectado no primeiro *flip-flop*, cuja saída é o *bit* menos significativo (**LSB** – *Least Significant Bit*). A saída do último *flip-flop* é o *bit* mais significativo (**MSB** – *Most Significant Bit*).

4.2 Detalhamento Conceitual

Considerar-se-á o *flip-flop* tipo D nas análises seguintes.

O *flip-flop* tipo D, essencialmente, copia o dado da entrada para a saída na ocorrência de uma borda do pulso de *clock*. No caso, será utilizado um sensível à borda de descida, conforme ilustrado pela figura 2, a qual também possui um exemplo de operação apresentado na carta de tempos.

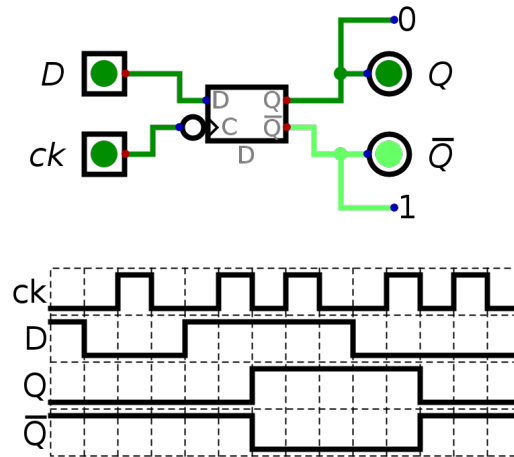


Figura 2: Flip-Flop tipo D e exemplo de operação.

Ao se *realimentar* o \bar{Q} na entrada D , a cada pulso de *clock*, tem-se a inversão do sinal de saída (ver figura 3).

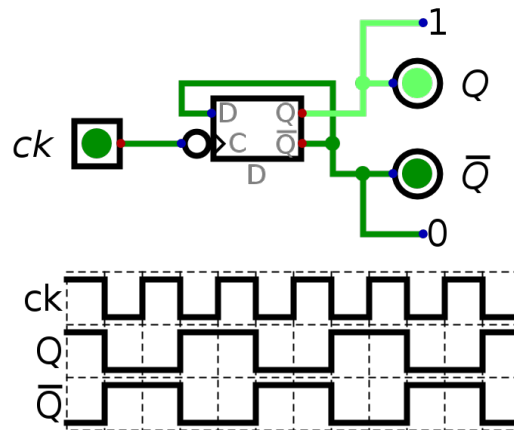


Figura 3: Flip-Flop com realimentação e exemplo de operação.

Os *flip-flops* possuem também sinais de *set* e *reset*, permitindo definir a saída principal (Q) em 1 ou 0, respectivamente (ver figura 4).

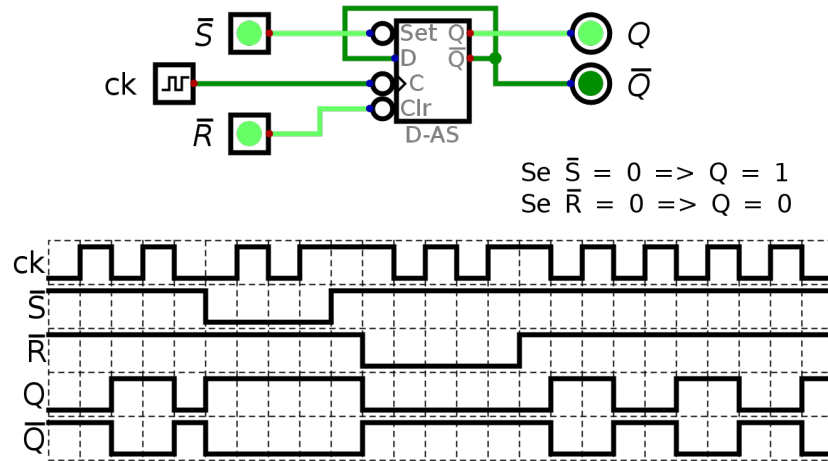


Figura 4: Flip-Flop tipo D com *set* e *reset* e exemplo de operação.

Associando-se dois conjuntos como este, tem-se que o *clock* do segundo *flip-flop* será fornecido pela saída Q do primeiro. Note-se, na figura 5, que ocorre a alteração de Q_1 nas bordas de descida de Q_0 . Q_1 é o bit mais significativo, por alterar menos frequentemente que Q_0 .

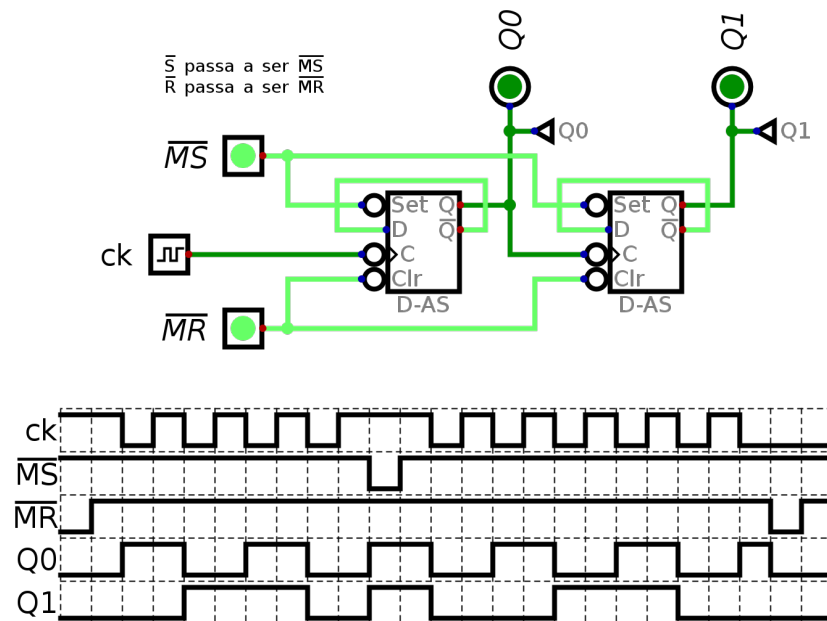


Figura 5: Contador de 2 bits e exemplo de operação.

Agrupando-se, agora, quatro estágios semelhantes, tem-se um contador

de 4 bits de 0 a 15 (ver figura 6). Note-se que Q_3 é o bit mais significativo.

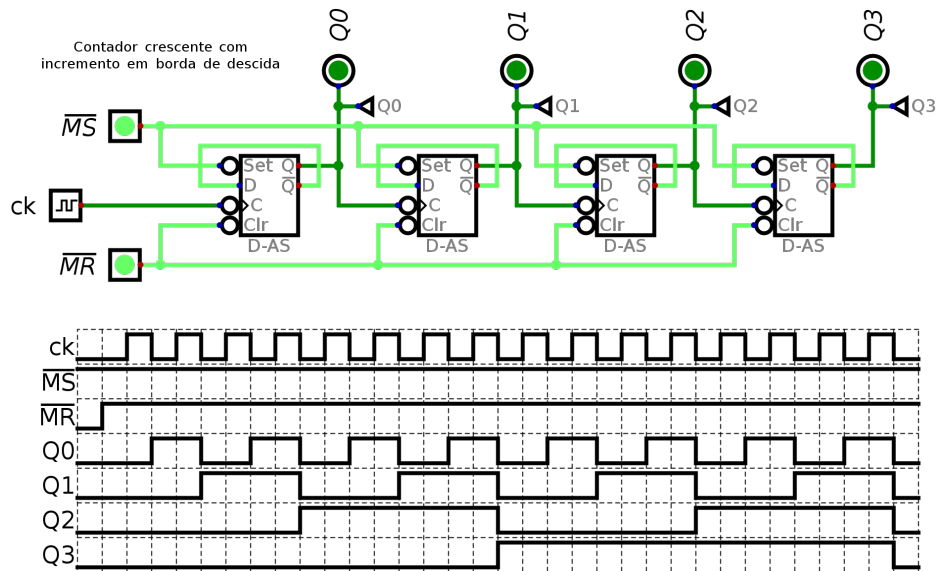


Figura 6: Contador assíncrono de 4 bits de 0 a 15 e exemplo de operação.

A borda do *clock* pode alterar o comportamento da contagem. Se os *flip-flops* forem de borda de subida, a contagem será decrescente (ver figura 7).

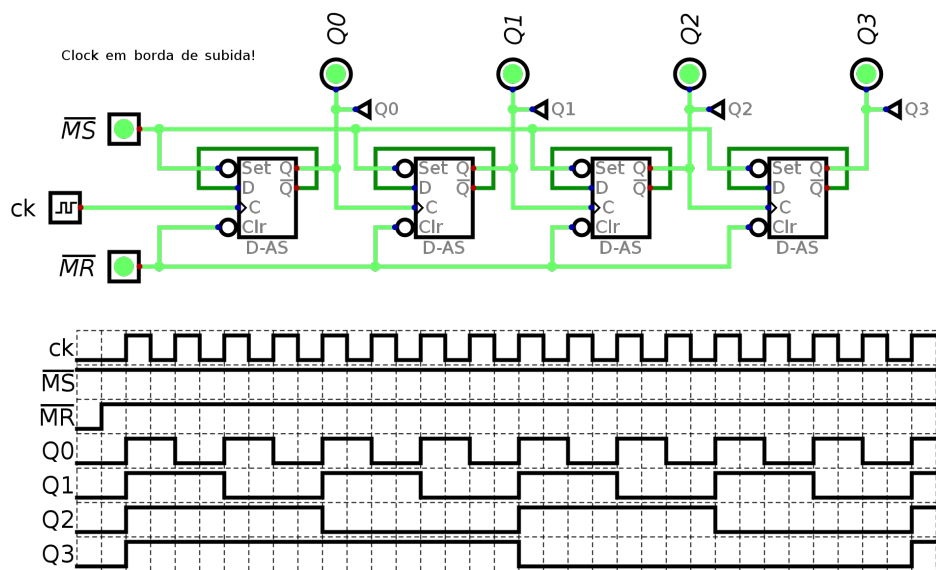


Figura 7: Contador decrescente e exemplo de operação.

Utilizando-se as saídas \bar{Q} para alimentar o *clock* do *flip-flop* seguinte, tem-se o equivalente acionamento por borda de descida e, portanto, a contagem passa a ser crescente (ver figura 8).

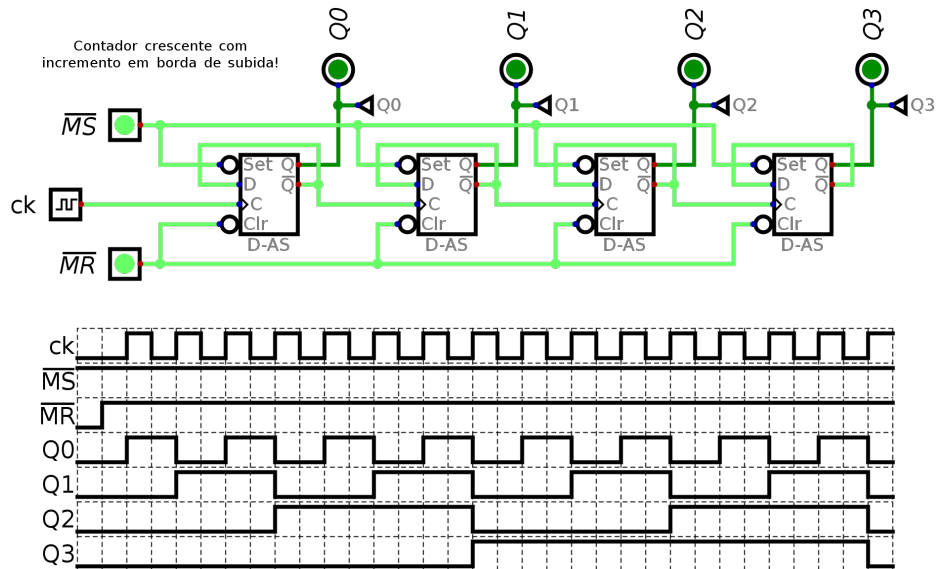


Figura 8: Contador assíncrono crescente de 0 a 15 com incrementos nas bordas de subida do *clock*.

Para se realizar uma contagem de 0 a 9, deve-se forçar o *reset* antes de se completar todo o ciclo do contador, ou seja, antes de ultrapassar o 15 e retornar a zero. Utilizando-se um circuito lógico combinatório, pode-se facilmente provocar tal efeito. O *reset* deve ocorrer assim que a contagem atingir 10 (ver figura 9).

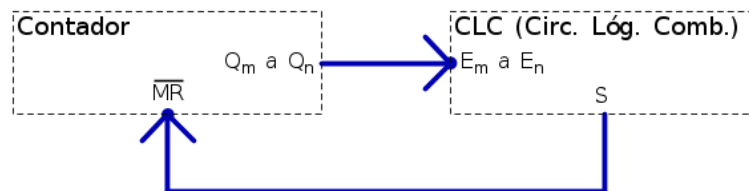


Figura 9: Blocos que ilustram a realimentação para se forçar o *reset* no estado desejado.

Finalmente, a implementação a seguir (figura 10) provoca o *reset* no momento em que a contagem atinge 10, garantindo, assim, um período de *clock*

The circuit diagram shows a 4-bit counter implemented with four D-AS flip-flops. The clock (ck) is connected to the clock input (C) of all flip-flops. The master reset (MR) signal is connected to the clear (Clr) input of all flip-flops and to a 4-input AND gate. The AND gate inputs are the outputs Q3 and Q1. The output of the AND gate is connected to the clear (Clr) input of the flip-flop whose Q output is Q0. The data inputs (D) are connected as follows: D0 = Q3, D1 = Q2, D2 = Q1, and D3 = Q0. The outputs are Q0, Q1, Q2, and Q3.

The timing diagram shows the waveforms for ck, MR, Q0, Q1, Q2, and Q3. The clock (ck) is a periodic square wave. The master reset (MR) is active-low, indicated by a bubble on the input symbol, and is high for most of the time. The outputs Q0, Q1, Q2, and Q3 show a binary count sequence from 0000 to 1111, with Q0 being the least significant bit and Q3 the most significant bit.

Figura 10: Contador assíncrono crescente de 0 a 9.

4.3 Algumas Aplicações

1. Relógios
2. Cronômetros
3. Contador de pulsos (de itens que passam por uma esteira, por exemplo)
4. Divisores de frequência

4.4 Sobre Divisores de Frequência

Todo contador, independentemente de ser assíncrono ou síncrono, é, por natureza, um divisor de frequência.

A divisão se observa ao comparar uma das saídas do contador com o sinal de *clock* da entrada. Assim, conforme o interesse da divisão a ser realizada, projeta-se o contador.

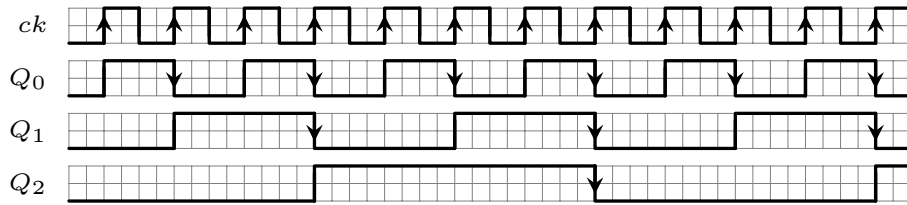


Figura 11: Carta de tempos para o contador da figura 1.

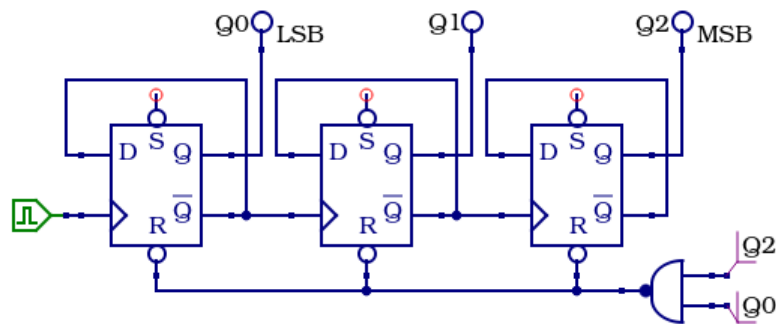


Figura 12: Contador de 0 a 4 (em binário, de 000_2 a 100_2). Observe-se que, ao ocorrer o avanço para o estado 5, rapidamente o contador retorna a 0. (Nota: considere-se o \bar{S} conectado em nível lógico “1”.)

Considerando-se o exemplo da figura 1, tem-se a carta de tempos da figura 11.

Na carta de tempos, percebe-se que Q_0 possui metade da frequência do *clock*, Q_1 possui metade da frequência de Q_0 e Q_2 metade da de Q_1 . Ainda, Q_1 possui $1/4$ da frequência do *clock* e Q_2 , $1/8$. Portanto, de acordo com a saída escolhida, tem-se disponível uma determinada divisão: $1/2$, $1/4$ ou $1/8$.

Num caso mais genérico, pode-se ter um contador não múltiplo da potência de 2, como o da figura 12, cuja carta de tempos está na figura 13.

Comparando-se Q_2 e *clock*, na carta de tempos, nitidamente observa-se a necessidade de 5 pulsos de *clock* para Q_2 dar um pulso. Em outras palavras, há 5 períodos de *clock* para 1 período de Q_2 . Portanto, o contador de 0 a 4 ilustrado é um divisor de frequência por 5.

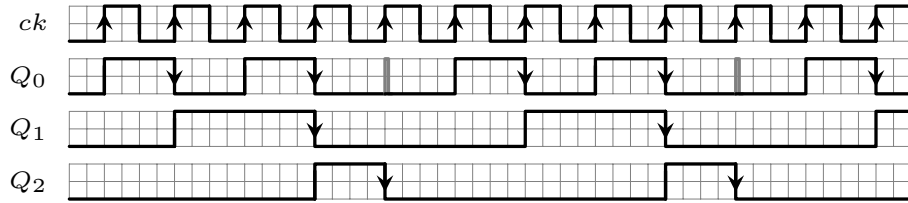
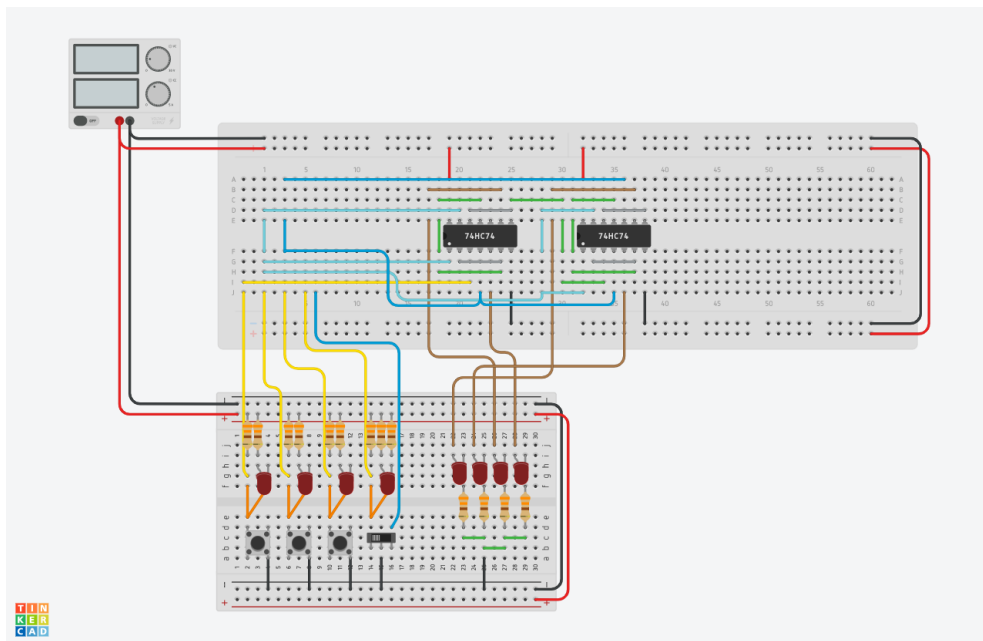


Figura 13: Carta de tempos para o contador da figura 12. Perceba-se que Q_0 apresenta um pulso com duração de alguns nanossegundos antes de ser zerado, no quinto pulso de *clock*.

5 Parte Experimental

5.1 Contador Assíncrono de 0 a 15

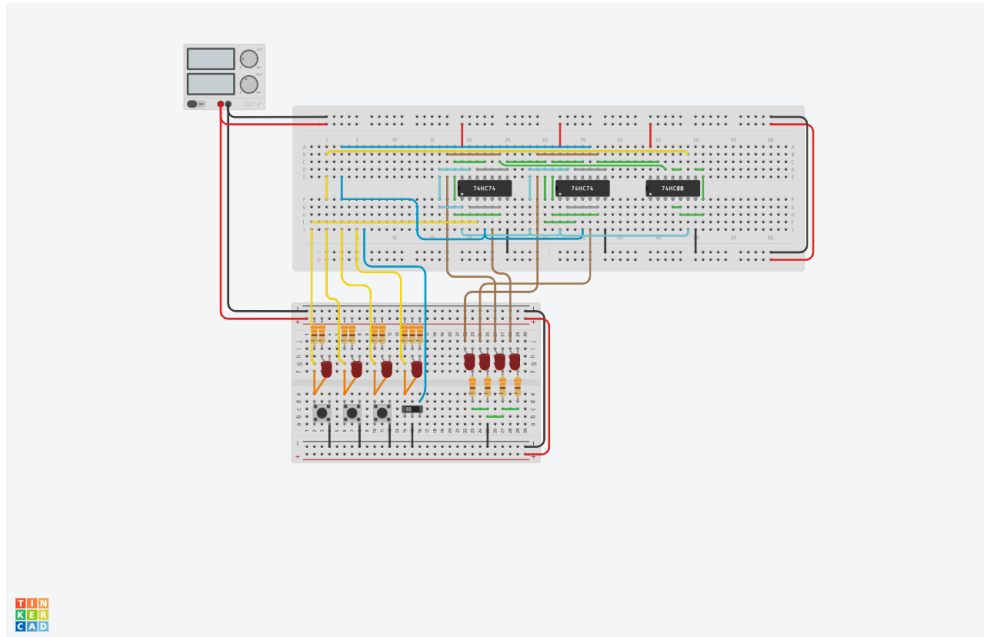
1. Carregar o circuito **ETE102-Contador_Assíncrono** no *Tinkercad*.



2. Observar a operação do \overline{MS} , \overline{MR} e do ck .
3. Em que momento ocorre o incremento do contador? Ao pressionar ou soltar o botão? Para tal, é gerada uma borda de subida ou de descida?

5.2 Contador Assíncrono de 0 a 9

1. Carregar o circuito **ETE102-Contador_Assíncrono_0_a_9** no *Tinercad*.



2. Observar a operação do \overline{MS} , \overline{MR} e do ck .
3. O que ocorre ao se desligar e ligar a fonte no simulador? O que ocorreria num circuito real?
4. Como garantir que a contagem sempre se inicie em um valor conhecido?
5. Qual o problema que ocorre com o \overline{MS} .

6 Entregas

1. Na montagem **ETE102-Contador_Assíncrono_0_a_9**, por que o \overline{MS} "não funciona"? Ou seja, por que, ao soltá-lo, todos os LEDs não ficam em 1?
2. Que alterações seriam necessárias no circuito para que, após aplicar \overline{MS} , tenha-se o número 9 (1001)?

3. Implementar um contador assíncrono crescente de 0 a 9 utilizando o CI 74HC93.

7 Opcionais para Exercitar a Imaginação

1. Implementar um contador crescente/decrecente de 0 a 15.
2. Implementar um contador crescente/decrecente automático de 0 a 15.
Ou seja, na contagem crescente, após atingir o 15, inicia-se a contagem decrescente e, após atingir o zero, inicia-se novamente a contagem crescente.
3. Alterar o circuito anterior para implementar um contador crescente/decrecente automático de 0 a 9.