

Sistemas Digitais

1º Ano de Engenharia Informática



Trabalho Prático n.º 8

Circuitos sequenciais: *Latches, Flip-Flops e Contadores*

Grupo

Diogo António Costa Medeiros n.º 70633

☐

_____ n.º _____

☐

_____ n.º _____

☐

Turma 5

Objectivos

- Construir uma *latch* $\bar{S} \bar{R}$ usando portas NAND
- Verificar experimentalmente as propriedades lógicas de um *flip-flop JK master-slave* (mestre-escravo)
- Determinar as ligações necessárias para um contador binário bidireccional (*up-down counter*)
- Construir contadores com sequência de contagem pré-estabelecida usando contadores bidireccionais.

Referências

- TAUB, Herbert, “Circuitos Digitais e Microprocessadores”, McGraw-Hill
- Texas Instruments online [<http://www.ti.com/>]

Material

- Placa RH21
 - 74LS00 — NAND, 2 entradas
 - 74LS20 — NAND, 4 entradas
 - 74LS76A — DUAL J-K FLIP-FLOPS, PRESET AND CLEAR
 - 74LS193 — SYNCHRONOUS UP/DOWN DUAL CLOCK COUNTER
 - 4 LEDs
 - 4 R = 330Ω
-

1. *Latch* e *Flip-Flop*

Flip-flop e *latches* são circuitos lógico fundamentais cuja saída é 1 ou 0, e que podem ser usados como **armazenadores de informação** binária, **divisores de frequência**, **contadores**, **detec- tores de sequências** binárias, etc. Como o seu **estado** se mantém por tempo indefinido, enquanto não for dada uma ordem explícita para a sua alteração, o *flip-flop* e a *latch* são classificados como **dispositivos bi-estáveis**.

Uma palavra se impõe no que toca à **nomenclatura**: Não existe um *standard* aceite por todos os fabricantes e autores neste assunto. Entre os dispositivos bi-estáveis temos a considerar os dispositivos **síncronos** e os **assíncronos**; os primeiros são facilmente identificáveis por possuírem uma entrada para o **signal de relógio de sincronismo**. Alguns autores classificam tudo como *flip-flop*, acrescentado a designação ‘**síncrono**’ ou ‘**assíncrono**’; outros preferem a designação *latch* (distinguindo também entre *latches* síncronas e assíncronas); outros, ainda, designam os dispositivos síncronos como *flip-flops* e os assíncronos como *latches*...

Neste protocolo usaremos esta última nomenclatura.

1.1 Latch $S'R'$

Uma latch $\bar{S} \bar{R}$ (também chamado *flip-flop* $\bar{S} \bar{R}$ básico) pode ser construído ligando duas portas NAND como mostra a figura 1.

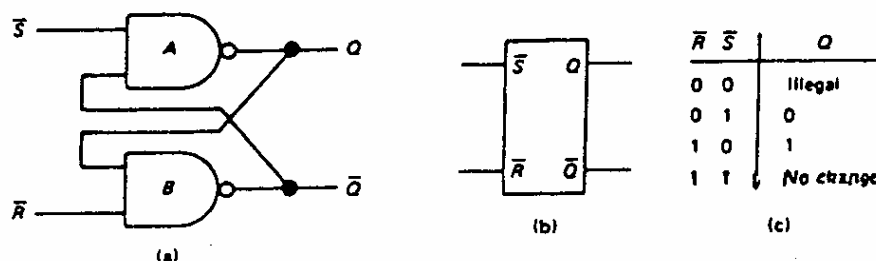


Figura 1. Latch $S'R'$ (a) Circuito lógico (b) Símbolo (c) Tabela de verdade.

1.1.1 Monte o circuito da figura 1 e complete a tabela ao lado.

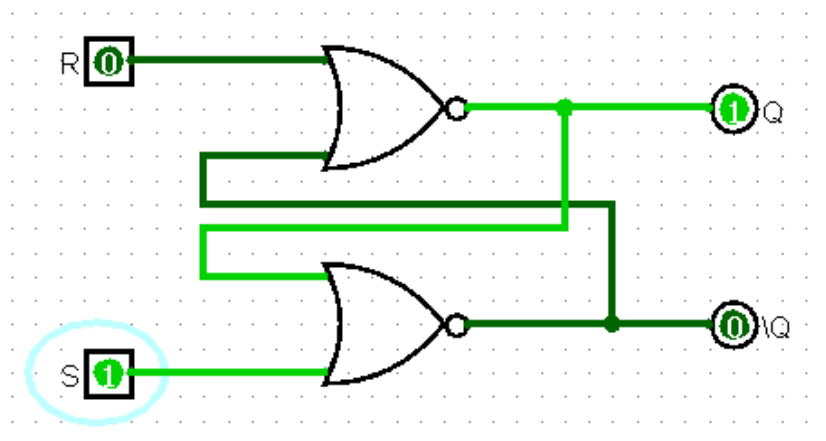
Atenção: Preencha a tabela ao lado **exactamente pela ordem indicada e sem desligar a alimentação** entre cada leitura, ou destruirá a *sequência* de valores de entrada!

\bar{S}	\bar{R}	Q	\bar{Q}
0	1	1	0
1	1	1	0
1	0	0	1
1	1	0	1
0	0	1	1

1.1.2 Comente os resultados obtidos.

\bar{S} e \bar{R} são entradas ativas-baixas. Quando \bar{S} está ativo e \bar{R} inativo, dá-se um Set das saídas, ficando Q a 1 e \bar{Q} a 0. Caso se desliguem ambas as entradas ($\bar{S}=\bar{R}=1$), as saídas permanecem com o último estado lógico, $Q_n=Q_{n-1}$ e $\bar{Q}_n=\bar{Q}_{n-1}$. Ao ligar-se \bar{R} a 0 e \bar{S} a 1, ocorre Reset, ficando $Q = 0$ e $\bar{Q} = 1$. Por último, com $\bar{S}=\bar{R}=0$, as saídas ficam ambas ativas, tratando-se de um estado inválido da latch.

1.1.3 Apresente o esquema de uma latch SR (implementada usando apenas portas NOR).



1.2 Flip-flop SR com relógio

Na figura 2 podemos ver um *flip-flop SR* com a entrada de relógio de sincronismo (*shift*) e a respectiva tabela de verdade.

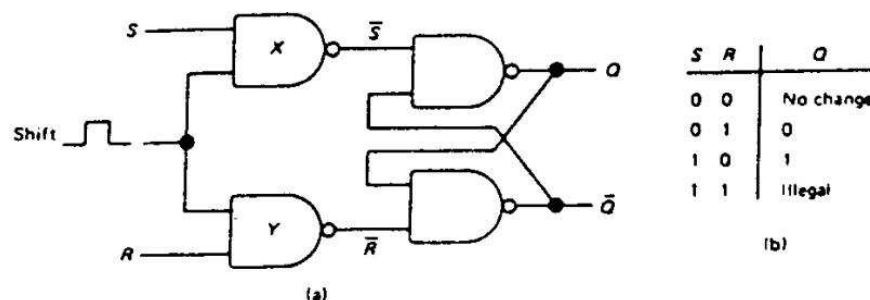


Figura 2. *Flip-flop SR* (a) Circuito lógico (b) Tabela de verdade.

1.2.1 Monte o circuito da figura 2. Use para *shift* uma onda quadrada entre 0 e 5V com 0.2s de período. Preencha a tabela ao lado.

Atenção: Preencha a tabela ao lado **exactamente pela ordem indicada e sem desligar a alimentação** entre cada leitura, ou destruirá a *sequência* de valores de entrada!

S	R	Q	Q̄
0	1	0	1
0	0	0	1
1	0	1	0
0	0	1	0
1	1	1	1

1.2.2 Comente os resultados obtidos.

As saídas do flip-flop SR atualizam quando se dá um pulso de relógio na borda de 0 para 1. Neste flip-flop é possível fazer Set das saídas com entradas S=1 e R=0 e Reset com entradas S=0 e R=1. Caso S=R=0, as saídas permanecem idênticas às anteriores no pulso seguinte. Já se S=R=1, as saídas ficam ambas ativas, resultando num estado inválido do circuito.

1.3 O Flip-flop JK master-slave

A maior parte dos sistemas digitais opera em **modo síncrono**, isto é, as acções são sincronizadas com um **relógio** do sistema, pelo que todos os *flip-flops* do circuito mudam (eventualmente) de estado ao mesmo tempo, em sincronismo com o relógio do sistema.

Como exemplo, temos o *flip-flop master-slave* (mestre-escravo) com relógio; o símbolo deste *flip-flop* é mostrado na figura 3, conjuntamente com a tabela de verdade.

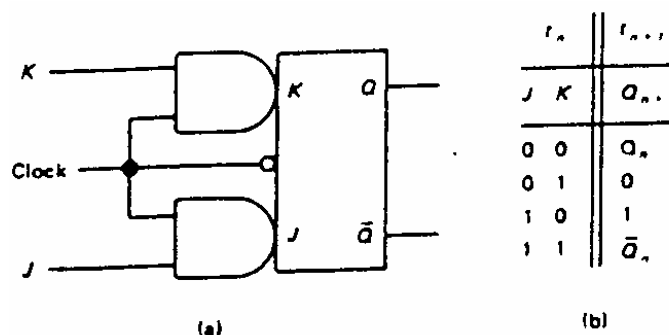


Figura 3. Flip-flop JK master-slave. (a) Símbolo (b) Tabela de verdade.

Na tabela de verdade, t_n é o **instante** imediatamente **anterior** ao pulso de relógio, e t_{n+1} é o **instante** imediatamente **posterior** à ocorrência desse pulso.

1.3.1 Examine as folhas de dados do *flip-flop JK* ('76). Monte o circuito da figura 3, aplicando-lhe o mesmo sinal de relógio do ponto 1.2.1 e complete a tabela de verdade.

J	K	Q	\bar{Q}
0	1	0	1
1	1	1	0
1	0	1	0
1	1	0	1
0	0	0	1

Atenção: Preencha a tabela ao lado **exatamente pela ordem indicada e sem desligar a alimentação** entre cada leitura, ou destruirá a *sequência* de valores de entrada!

1.3.2 Comente os resultados obtidos.

No circuito pedido, as saídas atualizam quando o pulso do relógio transita de 1 para 0 (borda de descida). Este circuito funciona de forma semelhante ao flip-flop SR, permitindo fazer Set com $J=1$ e $K=0$, Reset com $J=0$ e $K=1$ e manter os valores lógicos das entradas anteriores $J=K=0$. No entanto, no caso $J=K=1$, a cada pulso de relógio, as saídas invertem, $Q_n = \bar{Q}_{n-1}$ e $\bar{Q}_n = Q_{n-1}$.

1.3.3 Deixe as entradas J e K do '76 em aberto ($\equiv '1'$) e aumente a **frequência** do relógio para **1 kHz**. O resultado é um *flip-flop T*, como mostrado na figura 4.

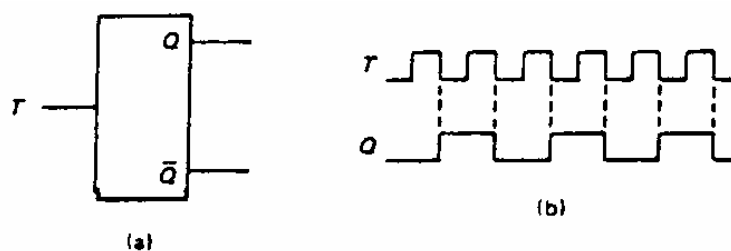


Figura 4. Flip-flop T. (a) Símbolo (b) Tabela de verdade.

1.3.4 Note que se trata de um **divisor de frequência**.

Indique o valor dessa divisão: 2 .

1.3.5 Com o *flip-flop* ainda ligado para funcionar no tipo *T*, aplique um nível 0 à entrada \overline{PRE} e depois a \overline{CLR} para verificar que estas entradas fazem, respectivamente, o *set* e o *reset* **directos** (também ditos **assíncronos**) do *flip-flop*.

2. Contadores

Os contadores binários são usados em aplicações que requerem uma evolução **crescente** da contagem binária. No entanto, é por vezes desejável usar um contador que progrida na contagem de forma **decrecente**. Um contador que possa ser usado em contagem crescente ou decrescente é designado *up-down counter* (por vezes, também ‘**contador bidireccional**’). O ’193 é um *up-down counter* síncrono de 4 bits. Tem uma entrada de *reset* directo (*CLR*), pelo que pode fazer um *reset* em qualquer ponto da contagem, independentemente do sinal de sincronismo. A entrada de controlo \overline{LOAD} permite carregar dados nas entradas *A*, *B*, *C* e *D*. O símbolo lógico deste contador é mostrado na figura 5 (página seguinte).

2.1 Examine as folhas de dados do ’193. Ligue \overline{LOAD} a V_{CC} e *CLR* à terra. Para **contagem crescente**, ligue *DOWN* a V_{CC} e aplique um sinal de relógio a *UP*. Use uma frequência baixa e verifique, usando LEDs (coloque resistências de 330Ω), que se trata de uma contagem crescente.

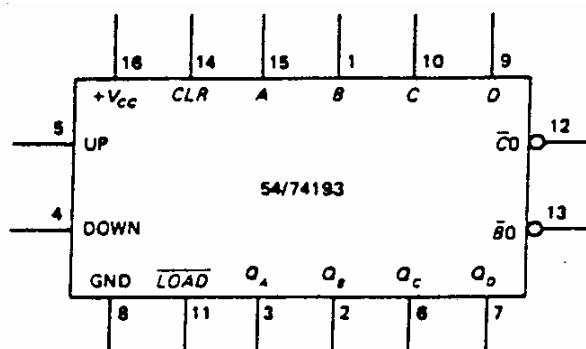


Figura 5. Símbolo lógico do ’193 (*4-bit binary up-down counter*).

2.2 Coloque agora *UP* a V_{CC} e aplique o sinal de relógio a *DOWN*. Verifique que obtém uma **contagem decrescente**.

2.3 Ligue o '193 como na figura 6. Ponha as quatro entradas a 0V, escolha a contagem crescente e verifique se os LEDs passam pelos 15 estados discretos.

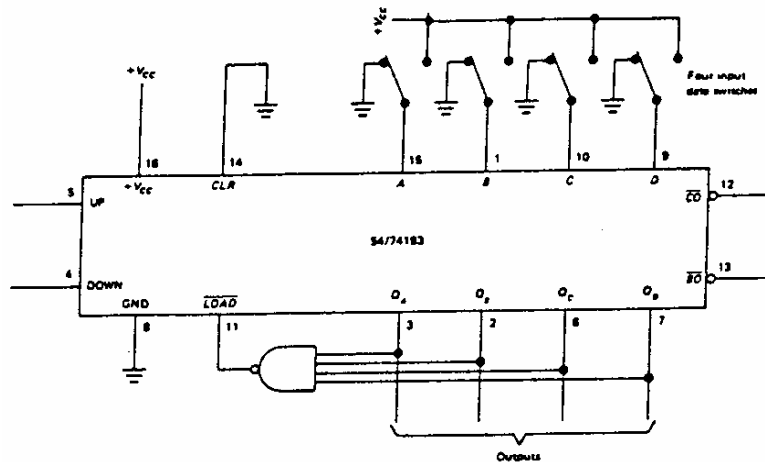
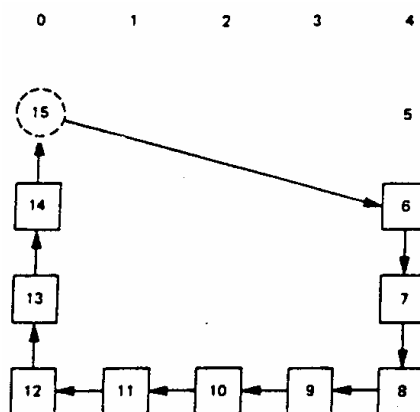


Figura 6. Circuito a implementar.

2.4 Coloque o circuito a funcionar de acordo com o seguinte **diagrama de estados**.



2.5 Teça as **considerações** que achar convenientes.

O circuito 74LS193 permite fazer contagens crescentes e decrescentes com início e fim em quaisquer números binários de 0 a 15. No exercício 2.4, montou-se este circuito de modo a fazer uma contagem de 6 a 14. Para isto, ligou-se DOWN a Vcc e UP ao clock, de modo a realizar-se uma contagem crescente. Para que a contagem tivesse início em 6, foi necessário colocar nas entradas D a A o número 6 em binário, 0110. Por outro lado, para que a contagem terminasse em 14, ligaram-se as entradas QD a QA a uma porta NAND e esta ao /LOAD, a fim de carregar 6 quando o contador atingisse 15, ainda que este último valor de saída não se verifique quando este circuito é ligado a outros circuitos, por exemplo um decodificador.