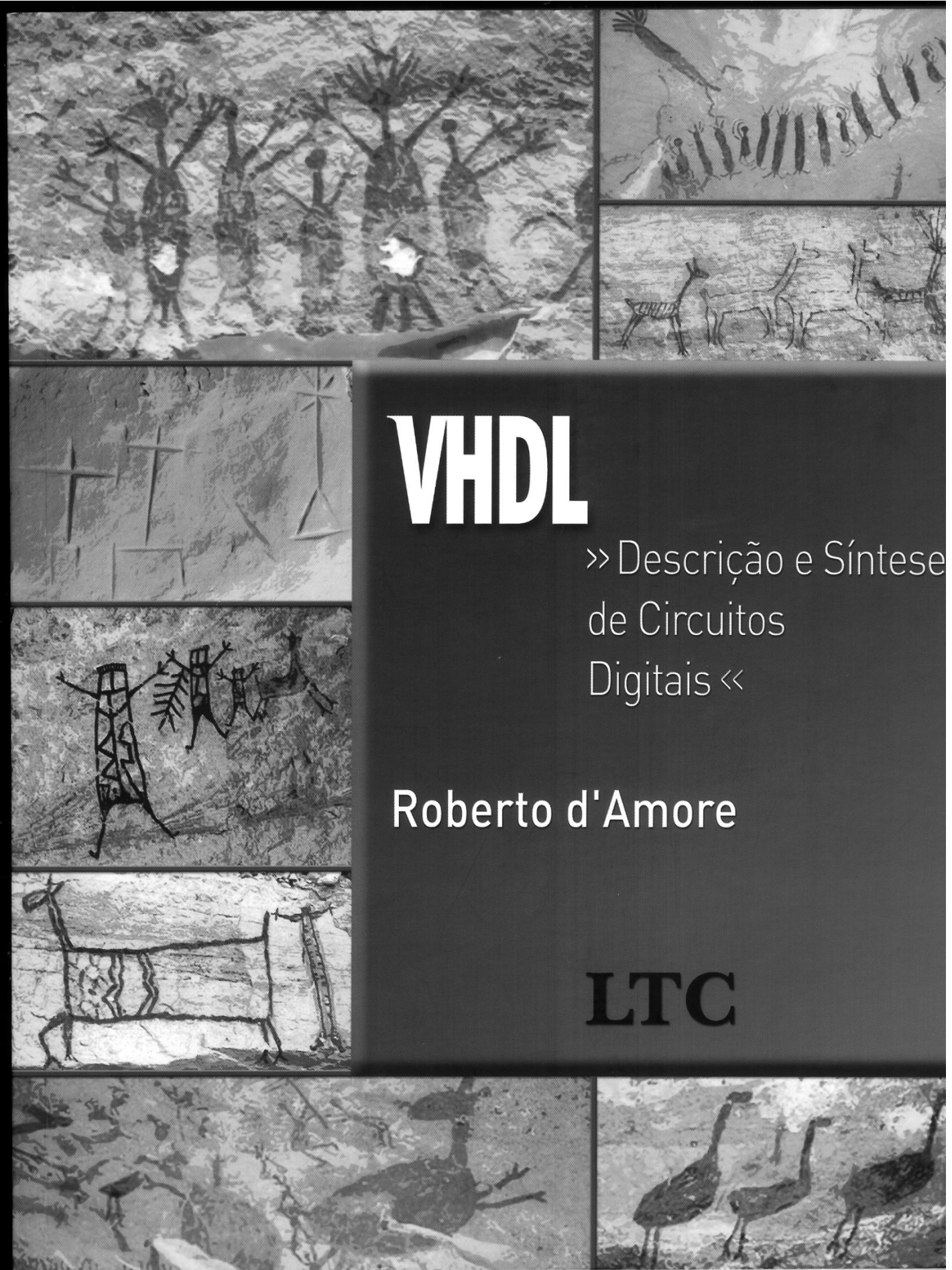


Lista de Exercícios – SEL0632

Capítulo 06



VHDL

» Descrição e Síntese
de Circuitos
Digitais «

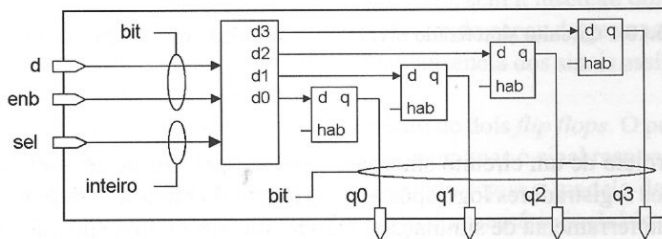
Roberto d'Amore

LTC

- Máquinas de estados podem ser descritas de modo semelhante a registradores. Uma construção “CASE WHEN” define transição de estados, e uma construção “IF ELSE” controla a transição de estados.
- A definição de um tipo enumerado é um modo prático para a descrição dos estados de uma máquina. Cada valor definido no tipo corresponde a um dos estados da máquina. As ferramentas de síntese suportam esse tipo de descrição, e a atribuição dos códigos para cada estado é realizada no momento da síntese. Diferentes modos de codificação são disponíveis nas ferramentas.
- Registradores podem ser inferidos através de sinais ou variáveis. Para a inferência de registradores com variáveis, é necessário que ela seja empregada antes de receber um novo valor.
- Sinais de inicialização assíncrona devem estar, obrigatoriamente, incluídos na lista de sensibilidade do processo. A não-inclusão leva a um comportamento não corretamente sintetizado pelas ferramentas de síntese.
- Circuitos síncronos devem ter um ciclo de inicialização. A falta desse procedimento pode levar a condições não previstas na simulação, pois o conteúdo dos registradores implementados é indeterminado no momento da energização do circuito.

6.12 Exercícios

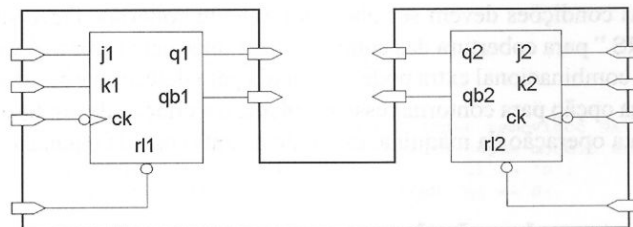
6.12.1 Apresente a descrição de um registrador sensível a nível de quatro bits. A entrada dos dados é feita por uma única entrada “d”. A seleção do elemento de memória é feita pela entrada “sel” do tipo inteiro. Uma vez selecionado o elemento, a saída correspondente segue a entrada “d” com “enb = 1”. A informação é mantida nas saídas com “enb = 0”. A Figura 6.12.1 ilustra o circuito proposto.



sel	enb	d	q0	q1	q2	q3
X	0	X	q0	q1	q2	q3
0	1	d _a	d _a	q1	q2	q3
1	1	d _b	q0	d _b	q2	q3
2	1	d _c	q0	q1	d _c	q3
3	1	d _d	q0	q1	q2	d _d

Figura 6.12.1 Ilustração para o Exercício 6.12.1.

6.12.2 Apresente a descrição de um *flip flop* tipo JK equivalente ao integrado 7473 da família TTL. A entrada “r1” realiza uma operação de *reset* assíncrona, a entrada “ck” é o sinal de relógio sensível a borda de descida, e as entradas “j” e “k” definem a operação no *flip flop*. Todas as entradas e saídas são do tipo “bit”. Veja a Figura 6.12.2.



	r1	ck	j	k	q	qb
reset ass.	0	x	x	x	0	1
troca	1	↘	1	1	q	qb
reset	1	↘	0	1	0	1
set	1	↘	1	0	1	0
mantém	1	↘	0	0	q	qb

Figura 6.12.2 Ilustração para o Exercício 6.12.2.

6.12.3 Apresente a descrição de um registrador com 4 bits e duas entradas de 4 bits, “da” e “db”. O sinal de seleção “sel” define de qual entrada o dado será armazenado na borda de descida do sinal de relógio “ck”. A operação do circuito é similar ao registrador 74298 da família TTL. Veja a Figura 6.12.3.

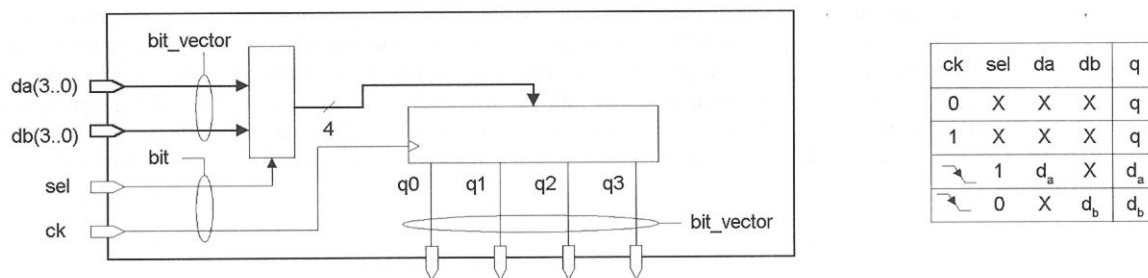


Figura 6.12.3 Ilustração para o Exercício 6.12.3.

6.12.4 Apresente a descrição de um contador com 4 bits crescente/decrescente com opção para carga de dados paralela e *reset* assíncrono. O contador deve contar entre os valores “0” a “11”. A Figura 6.12.4 ilustra o contador. O terminal “ld” permite o armazenamento do dado “d” na borda de subida do sinal de relógio “ck”. O terminal “crs” define se a contagem é crescente ou decrescente, e “rst” executa uma operação de *reset* assíncrona. Os sinais “d” e “q” são do tipo inteiro, e os sinais restantes do tipo “bit”.

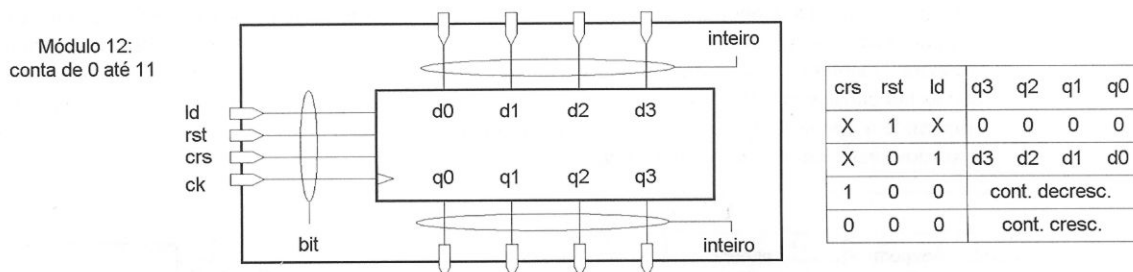


Figura 6.12.4 Ilustração para o Exercício 6.12.4.

6.12.5 Apresente a descrição de um contador decimal equivalente ao circuito integrado da família TTL tipo 7490. A tabela contendo os modos de operação do componente está apresentada na Figura 6.12.5. Os terminais “rst1” e “rst2” levam as saídas ao nível lógico “0000”, e os terminais “set1” e “set2” levam as saídas ao nível lógico “1001”. Ambas as operações são executadas de maneira assíncrona. O circuito é composto por um contador módulo 5 e um *flip flop*; a contagem módulo 10 é atingida com uma ligação pseudo-síncrona entre o *flip flop* e o sinal de relógio do contador módulo 5. A saída do contador deve ser do tipo “BIT_VECTOR”, e os demais sinais são do tipo “bit”.

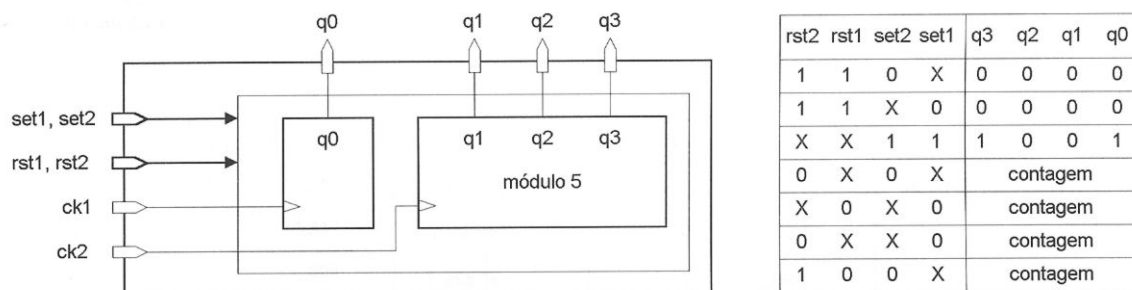


Figura 6.12.5 Ilustração para o Exercício 6.12.5.

6.12.6 Apresente a descrição de um registrador de deslocamento de 4 bits. A entrada serial dos dados é feita pelo terminal “de”, e a saída paralela dos dados é feita pelos terminais “qs(3)”, “qs(2)”, “qs(1)” e “qs(0)”. O deslocamento é processado na borda de subida do sinal “ck”. Empregue variáveis para armazenar os dados.

6.12.7 Apresente a descrição de um registrador de deslocamento equivalente ao circuito integrado 74166 da família TTL. O registrador contém 8 bits, possibilidade de carga paralela através das entradas “d0” a “d7” e uma entrada serial, terminal “ds”. O terminal “rst_l” leva a saída de todos os registradores ao nível lógico baixo de modo assíncrono. O terminal “ld_l” permite a carga dos dados presentes nas entradas de “d0” a “d7” na borda de subida do sinal “ck”. A operação de deslocamento é executada, também, na borda de subida do sinal “ck”. O terminal “ce_l” habilita as operações síncronas: deslocamento e carga paralela. A Figura 6.12.6 ilustra simplificada o componente e uma tabela contendo os modos de operação do registrador.

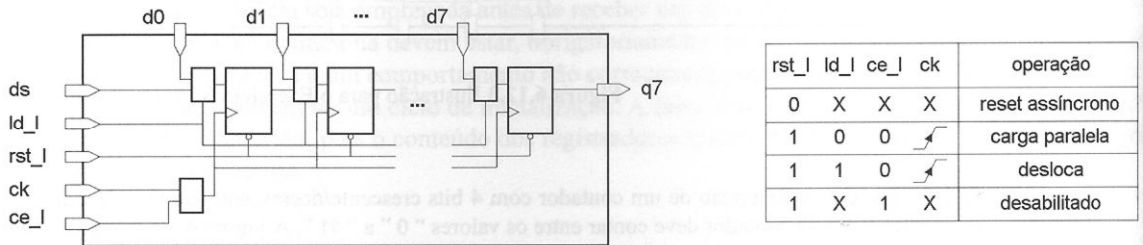


Figura 6.12.6 Ilustração para o Exercício 6.12.7.

6.12.8 Projete um controlador de semáforo para trânsito composto por três lâmpadas: vermelha, amarela e verde. O sinal “tempo” fornece a temporização do sistema, de modo que a cada pulso desse sinal é feita a transição para um novo estado. O sinal “início” leva o semáforo imediatamente para o estado inicial, causando o acendimento da lâmpada vermelha. O sinal “espera” mantém o estado atual, nos casos em que a lâmpada vermelha ou a lâmpada verde estão acesas. Esse sinal permite que as lâmpadas vermelha e verde fiquem acesas por um período maior de tempo. O sinal “noite” pisca a lâmpada amarela simulando a operação de um semáforo na madrugada. O diagrama de estados completo é apresentado na Figura 6.12.7. Os tipos dos sinais são indicados na figura.

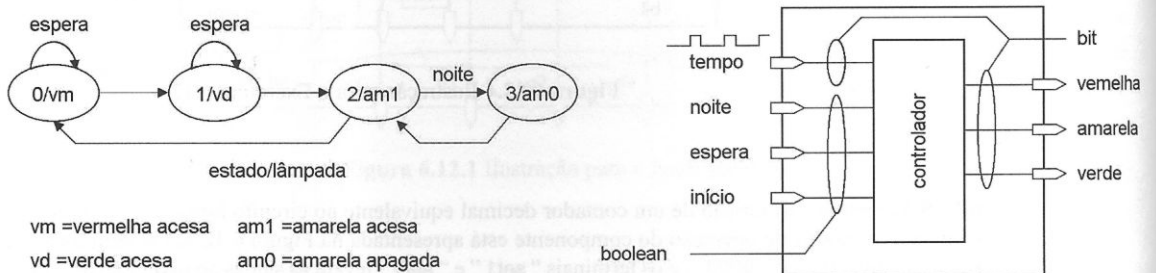


Figura 6.12.7 Ilustração para o Exercício 6.12.8.

6.12.9 Projete um controlador de endereços para uma memória RAM dinâmica. A memória contém 8 entradas para seleção de endereço e duas linhas denominadas “RAS” e “CAS” para controle. O controlador, para executar um ciclo de endereçamento, deve seguir as operações descritas na Figura 6.12.8. O sinal “ce_l=0”, na borda de subida da base de tempo “ck”,

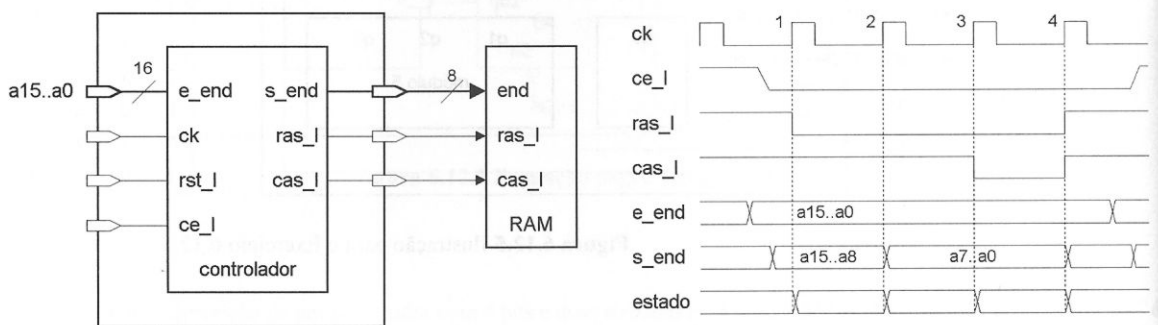


Figura 6.12.8 Ilustração para o Exercício 6.12.9.

inicia o ciclo ativando o sinal de seleção de coluna “*ras_l*”. Nesse estado, as linhas de endereço “*s_end*” correspondem aos 8 bits mais significativos das linhas de endereço da entrada. No segundo ciclo de relógio, as linhas “*s_end*” recebem os 8 bits menos significativos das linhas de endereço da entrada. No terceiro ciclo, o sinal “*cas_l*” é ativado, e no quarto ciclo de relógio a operação é terminada com os sinais “*ral_l*” e “*cas_l*” desativados.

6.12.10 O Quadro 6.7.1 contém uma descrição para inferência de *flip flops* empregando variáveis, entidade “*flip_1v*”. Altere a descrição de modo que os elementos inferidos sejam *latches*, em vez de *flip flops*.

6.12.11 Apresente o código para descrever o comportamento da máquina de estados ilustrada na Figura 6.12.9. Note que esta é uma máquina *Mealy*, onde o valor da saída depende do valor da entrada.

6.12.12 Sintetize a descrição proposta para o Exercício 6.12.11. Imponha para a ferramenta de síntese uma codificação binária para a atribuição de estados. Verifique o resultado, e observe o número de *flip flops* empregado para armazenar os estados.

6.12.13 Repita o Exercício 6.12.12 impondo, agora, uma codificação “único um” na atribuição de estados. Verifique o resultado, e compare o número de *flip flops* empregado para armazenar os estados com relação ao exercício anterior.

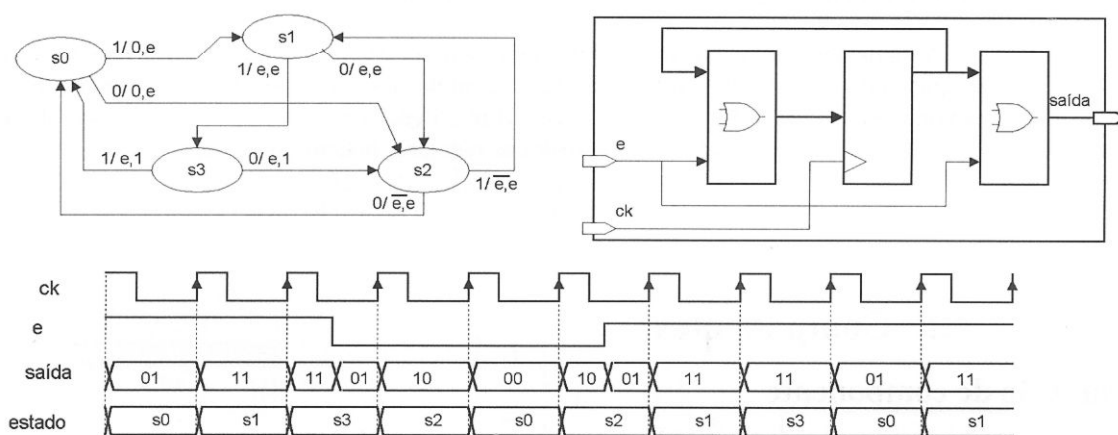


Figura 6.12.9 Ilustração para o Exercício 6.12.11.