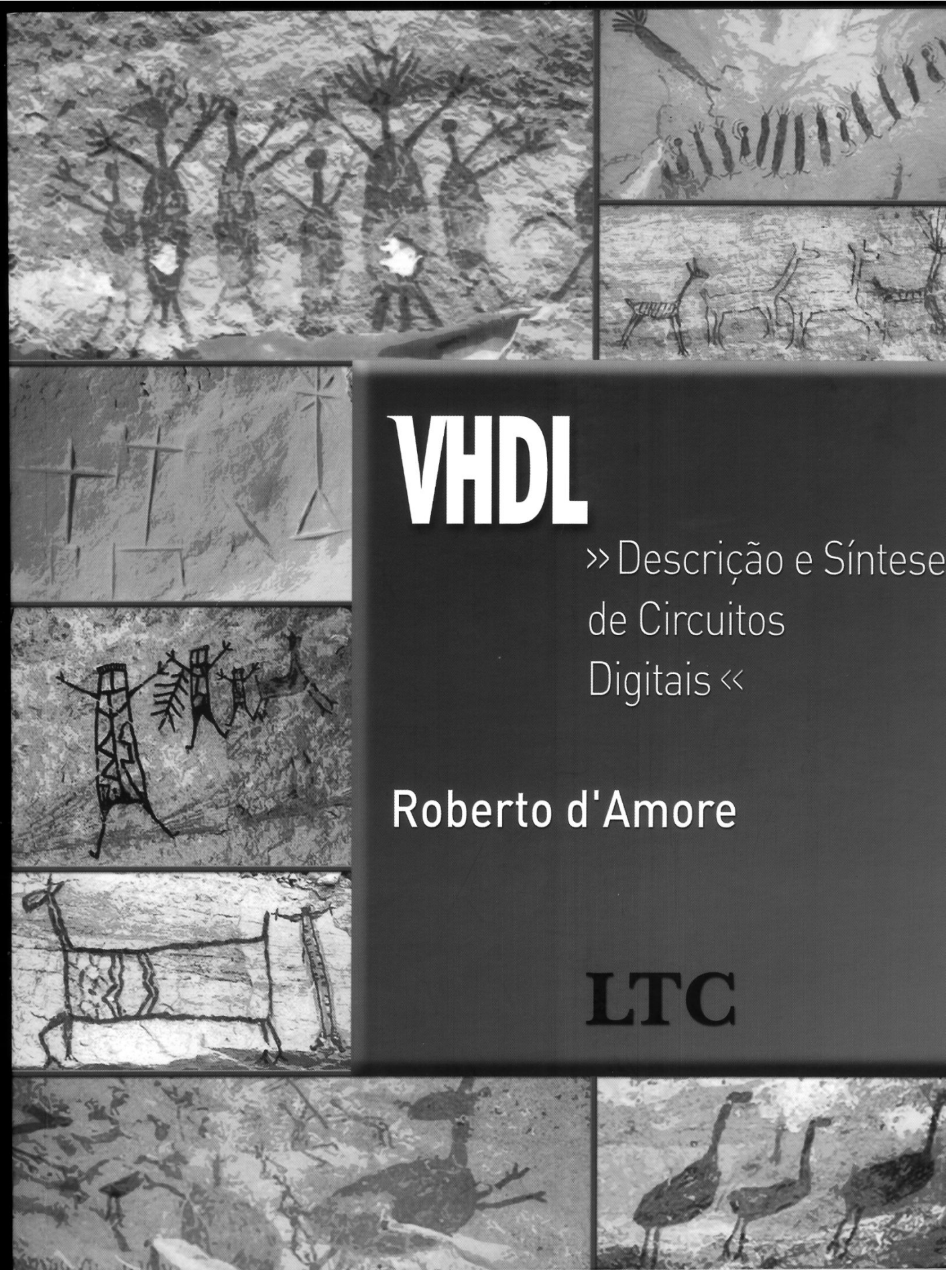


# Lista de Exercícios – SEL0632

## Capítulo 03



# VHDL

» Descrição e Síntese  
de Circuitos  
Digitais «

Roberto d'Amore

**LTC**

```

1 ENTITY latch_0a IS
2   PORT (enb, d1, d2, d3 : IN      BIT;
3         q1, q2, q3      : BUFFER BIT);
4 END latch_0a;
5
6 ARCHITECTURE teste OF latch_0a IS
7 BEGIN
8   q1 <= d1 WHEN enb = '1' ELSE
9     q1;
10
11  q2 <= d2 WHEN enb = '1';
12
13  WITH enb SELECT
14    q3 <= d3 WHEN '1',
15    q3 <= q3 WHEN '0';
16 END teste;

```

Quadro 3.8.3 Descrição de elementos de memória.

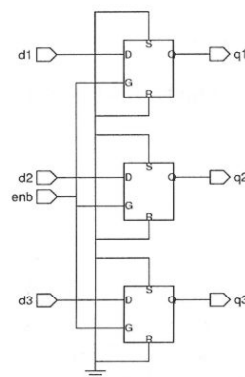


Figura 3.8.5 Circuito sintetizado, descrição “ latch\_0a ”, nível RTL.

### 3.9 Principais pontos abordados

- A execução dos comandos concorrentes por parte do simulador é possível pela introdução de um atraso interno nulo denominado  $\Delta$ . Um sinal somente recebe um novo valor após um intervalo de tempo igual a  $1\Delta$ .
- Na construção “ WHEN ELSE ”, a seqüência das condições define a prioridade, pois a primeira condição verdadeira encontrada na lista estabelece a expressão para o sinal de destino.
- Na construção “ WITH SELECT ” as condições devem ser mutuamente exclusivas; todas as condições devem ser especificadas, e elas possuem a mesma prioridade.
- O comando “ BLOCK ” permite isolar regiões de código concorrente. Sinais visíveis unicamente nessas regiões podem ser declarados.
- Sinais no interior de um bloco podem ser desabilitados conforme a condição de uma expressão de guarda. Esse tipo de construção pode ser empregado na inferência de um circuito com saída em alta impedância ou um elemento de memória.
- O comando “ PROCESS ” delimita as regiões contendo comandos seqüenciais. A execução dos comandos contidos em um processo pode ser condicionada a uma lista de sinais denominada lista de sensibilidade.

### 3.10 Exercícios

3.10.1 Proponha descrições para os circuitos da Figura 3.10.1. Analise a seqüência de iterações que devem ser executadas pelo simulador, com base nas formas de onda apresentadas na figura. Note que o atraso de propagação das portas é nulo.

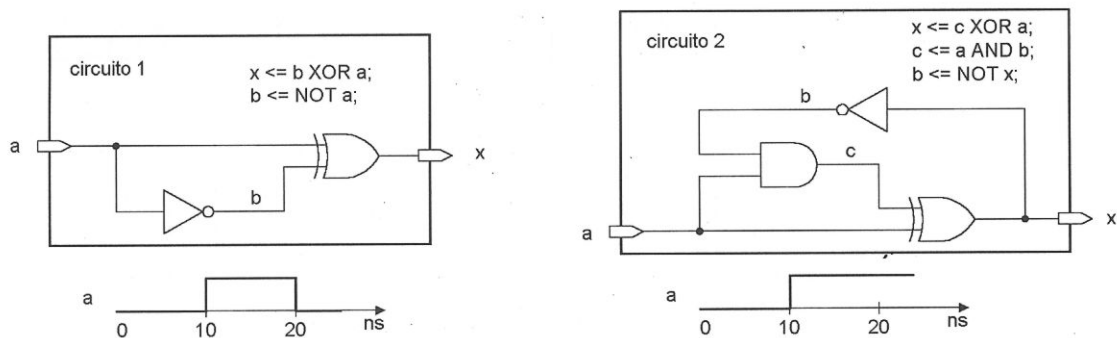


Figura 3.10.1 Exercício 3.10.1.

**3.10.2** Apresente o código de um circuito *latch* transparente de 4 bits, conforme ilustrado na Figura 3.10.2, empregando a construção “WHEN ELSE”. As saídas “q” seguem as entradas “d” na condição “en = 1”; na condição “en = 0” a informação é mantida.

**3.10.3** Repita o Exercício 3.10.2 empregando a construção “WITH SELECT”.

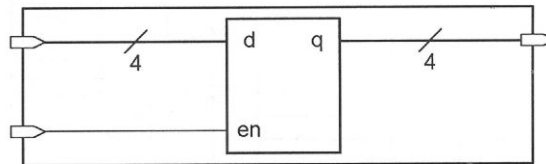


Figura 3.10.2 Ilustração para os Exercícios 3.10.2 e 3.10.3.

**3.10.4** Desenvolva o código para um detector de prioridade com quatro entradas empregando a construção “WHEN ELSE”. A entrada de prioridade máxima é “p0”, e a entrada de prioridade mínima é “p3”. O detector deve ser empregado num sistema microprocessado, onde quatro periféricos podem solicitar uma interrupção. Uma das entradas em nível alto corresponde a um pedido de interrupção; neste caso, a saída “int” deve ser ativada. Na Figura 3.10.3 é apresentada a tabela-verdade do decodificador.

**3.10.5** Repita o Exercício 3.10.4 empregando a construção “WITH SELECT”.

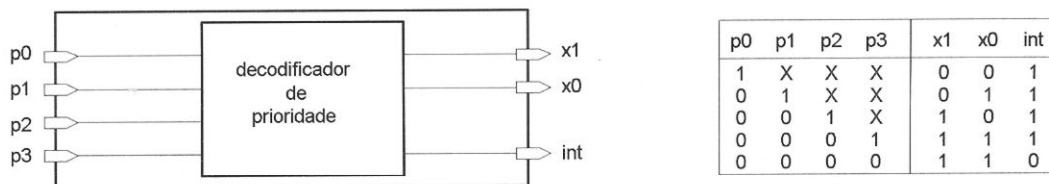


Figura 3.10.3 Ilustração para os Exercícios 3.10.4 e 3.10.5.

**3.10.6** Apresente o código de um circuito decodificador para a seleção de memórias. O sistema é composto por três memórias: a primeira respondendo na faixa de endereços de “00000 h” a “1FFFF h”, a segunda na faixa de “20000 h” a “27FFF h”, e a terceira na faixa de “28000 h” a “2FFFF h” (ver Figura 3.10.4). Empregue a construção “WHEN ELSE”.

**3.10.7** Repita o Exercício 3.10.6 empregando a construção “WITH SELECT”.

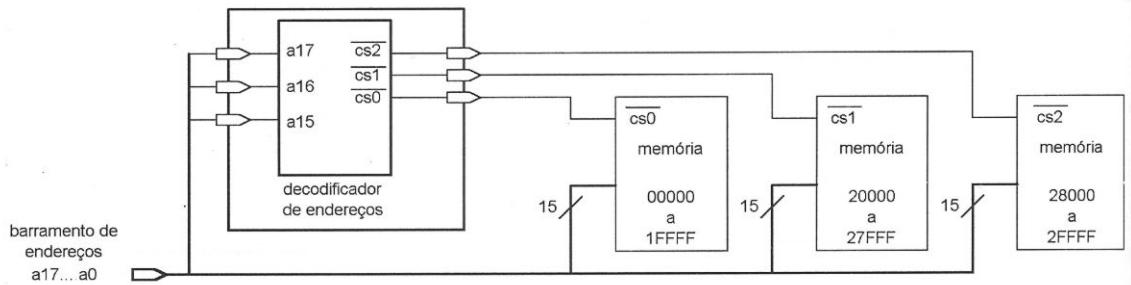


Figura 3.10.4 Ilustração para os Exercícios 3.10.6 e 3.10.7.

3.10.8 Apresente o código de um conversor do código BCD, *binary-coded decimal*, para um mostrador de sete segmentos. Os segmentos são acionados em nível lógico alto.

3.10.9 Com base na Figura 3.10.5, desenvolva o código descrevendo o sistema contendo um decodificador BCD para sete segmentos e um circuito de seleção. A descrição deve ser dividida em dois blocos, um contendo o decodificador, e o outro bloco, o circuito de seleção. Os mostradores são acionados em nível baixo, e os segmentos acionados em nível alto.

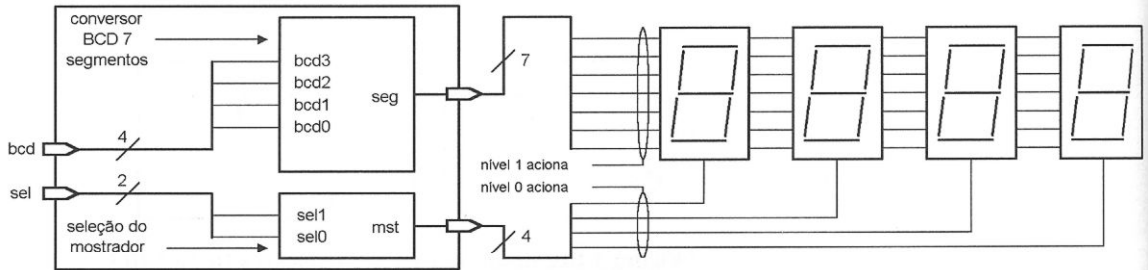


Figura 3.10.5 Ilustração para o Exercício 3.10.9.

3.10.10 Sintetize as descrições criadas para o registrador de quatro bits nos Exercícios 3.10.2 e 3.10.3 e compare os resultados no nível RTL.

3.10.11 Sintetize as descrições criadas para o detector de prioridade nos Exercícios 3.10.4 e 3.10.5 e compare os resultados. Verifique os diagramas gerados no nível RTL e no nível portas lógicas.

3.10.12 Sintetize as descrições criadas para o seletor de memória nos Exercícios 3.10.6 e 3.10.7 e compare os resultados. Verifique os diagramas gerados no nível RTL e no nível portas lógicas.