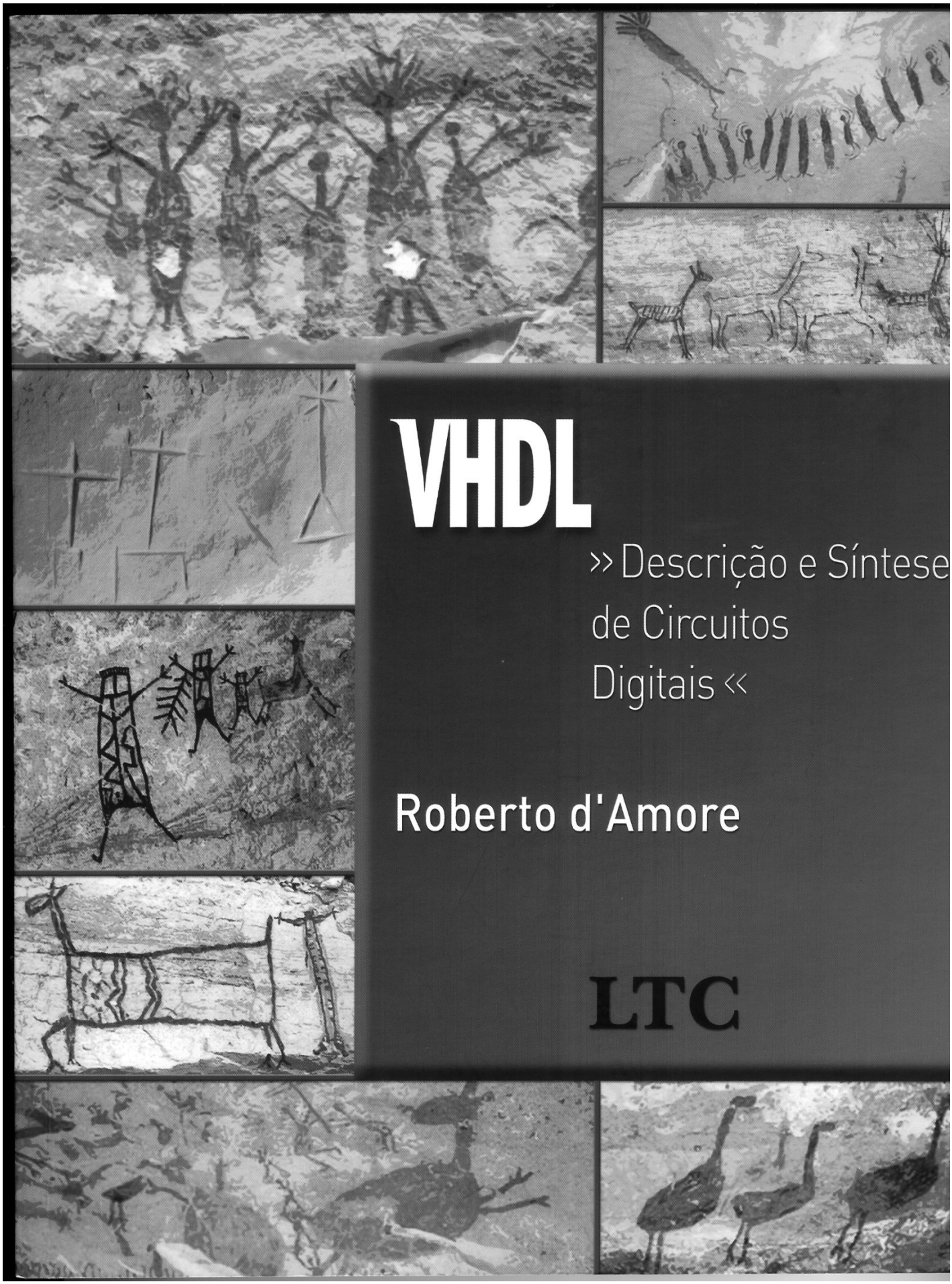
Lista de Exercícios – SEL0632

Capítulo 07



Na descrição do Quadro 7.4.11, a rotina de conversão de um valor contido num objeto do tipo vetor de bits para um tipo inteiro foi alterada para ilustrar o emprego do comando " EXIT ". Nesse caso, foi empregado um laço infinito, onde o fim da execução é definido pelo comando " EXIT " na linha 19.

```
1  ENTITY loop_e1 IS
2    GENERIC (n           : INTEGER := 4);
3    PORT (e_bit          : IN  BIT_VECTOR (n-1 DOWNTO 0); -- entrada tipo vetor de bits
4          inteiro_exit : OUT INTEGER RANGE 0 TO 2**n-1);-- saida convertida para inteiro
5  END loop_e1;
6
7  ARCHITECTURE teste OF loop_e1 IS
8
9  BEGIN
10   abc_for: PROCESS (e_bit)
11     VARIABLE temp, i : INTEGER;
12   BEGIN
13     temp := 0;
14     i := 0;
15     LOOP
16       IF e_bit(i) = '1' THEN temp := temp + 2**i;
17       END IF;
18       i := i +1;
19       EXIT WHEN i = e_bit'LENGTH;
20     END LOOP;
21     inteiro_exit <= temp;
22   END PROCESS;
23 END teste;
```

Quadro 7.4.11 Conversão " bit_vector " para o tipo " integer " - laço infinito.

A descrição do Quadro 7.4.11 submetida à ferramenta de síntese gerou uma mensagem de erro (ver Quadro 7.4.12). A ferramenta de síntese não suporta laços sem uma condição de teste, conforme apresentado na descrição.

```
"/vhdl/loop_e1.vhd",line 15: Error, infinite loops are not supported.
```

Quadro 7.4.12 Mensagem de erro: laço da descrição do Quadro 7.4.11 não é suportado.

7.5 Principais pontos abordados

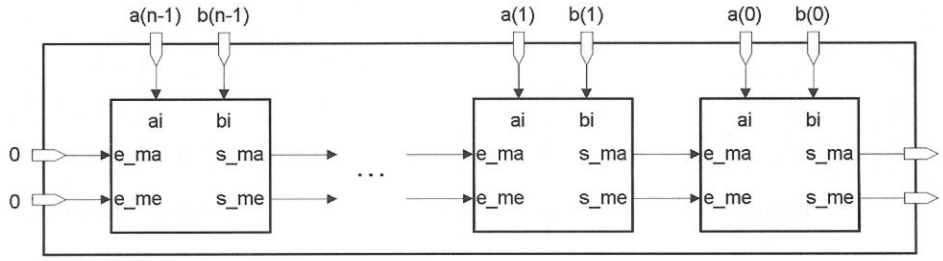
- Componente é uma entidade de projeto (declaração e arquitetura) empregada por uma arquitetura. Os componentes são empregados em projetos hierárquicos.
- A declaração de um componente é necessária para ele ser solicitado no interior de uma descrição. O formato da declaração é semelhante à declaração de uma entidade, onde são definidas as portas de entrada e saída.
- Na versão VHDL-1993 é permitida a solicitação de um componente sem uma correspondente declaração.
- Genéricos são empregados na transferência de informações estáticas para entidades de projeto e blocos.
- O comando " GENERATE " permite repetir comandos concorrentes. Dois esquemas de geração são possíveis: esquema " FOR " e esquema " IF ".
- O comando " LOOP " permite repetir a execução de comandos seqüenciais. Os esquemas de iteração disponíveis são: " FOR " e " WHILE ". A execução das iterações pode ser alterada pelos comandos " NEXT " e " EXIT ".

7.6 Exercícios

7.6.1 O objetivo do exercício é propor o código de um circuito comparador de dois números " a " e " b ". O circuito deve detectar se o valor de " a " é maior, menor ou igual ao valor de " b ". O número de bits de " a " e " b " é definido pelo genérico " n ".

O circuito comparador deve ser composto por células conforme ilustra a Figura 7.6.1. Cada célula possui quatro entradas " ai ", " bi ", " e_ma " e " e_me " e duas saídas " s_ma " e " s_me ". A saída " s_ma " em nível alto significa " a " maior

que " b "; a saída " s_me " em nível alto significa " a " menor que " b "; e as duas saídas em nível lógico baixo significa " a " igual a " b ". As entradas " e_ma " e " e_me " são interligadas à célula que avalia o próximo par de bits mais significativo. A comparação procede do par de bits mais significativo para o par de bits menos significativo. Apresente as descrições da célula e do comparador. O código do comparador deve solicitar a célula na forma de componente.



	s_ma	s_me
a > b	1	0
a < b	0	1
a = b	0	0

Figura 7.6.1 Diagrama de blocos do Exercício 7.6.1.

7.6.2 Sintetize a descrição proposta para o Exercício 7.6.1 no caso de um comparador com três bits. Observe o resultado da síntese, e compare os recursos empregados no componente que trata do bit mais significativo com os componentes restantes. Verifique se a ferramenta de síntese reduziu o número de componentes de alguma célula.

7.6.3 Altere a descrição proposta no Quadro 7.1.4 de modo a implementar um somador de quatro bits sem a entrada "vem um", e a saída "vai um", conforme ilustrado na Figura 7.6.2. Empregue a palavra reservada " OPEN " para os sinais desconectados dos componentes.

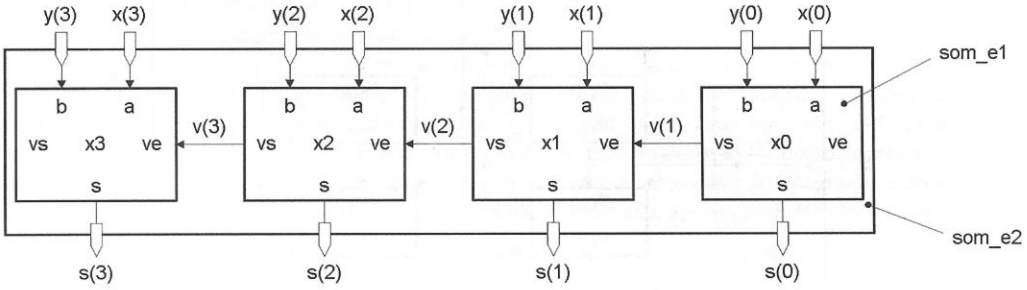


Figura 7.6.2 Somador de 4 bits sem entrada "vem um" e saída "vai um".

7.6.4 Sintetize a descrição proposta para o Exercício 7.6.3, e verifique se o resultado da síntese corresponde ao desejado. Verifique, também, como a ferramenta de síntese trata os blocos com sinais sem conexão. Ela elimina as portas lógicas desnecessárias nos componentes com sinais desconectados?

7.6.5 Apresente a descrição de um circuito contendo 3 contadores na forma de componentes. Os circuitos realizam contagens entre os valores 0 e 9, e a operação de *reset* é realizada de modo síncrono. As Figuras 7.6.3 e 7.6.4 ilustram a operação do conjunto. O valor na saída do contador é incrementado, se o sinal de habilitação " enb " estiver em nível alto, e o sinal " rst_s " estiver em nível baixo. A saída " cnt_9 " indica ao próximo contador que o final da contagem foi atingida, e este deve incrementar o seu valor no próximo ciclo de relógio. Observe, na Figura 7.6.4, que o sinal " cnt_9 " não pode apenas detectar o valor " 9 " do próprio contador. Considere o sinal de *reset*, " rst_s ", ativo em nível alto, todos os sinais do tipo bit, e as saídas " q " dos contadores como tipo inteiro.

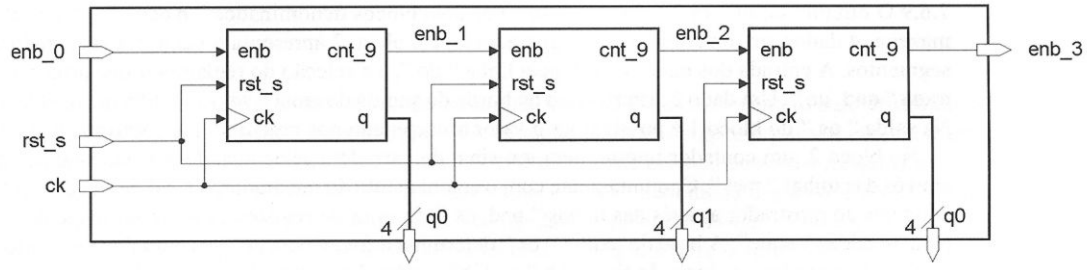


Figura 7.6.3 Contadores decimais síncronos - Exercício 7.6.5.

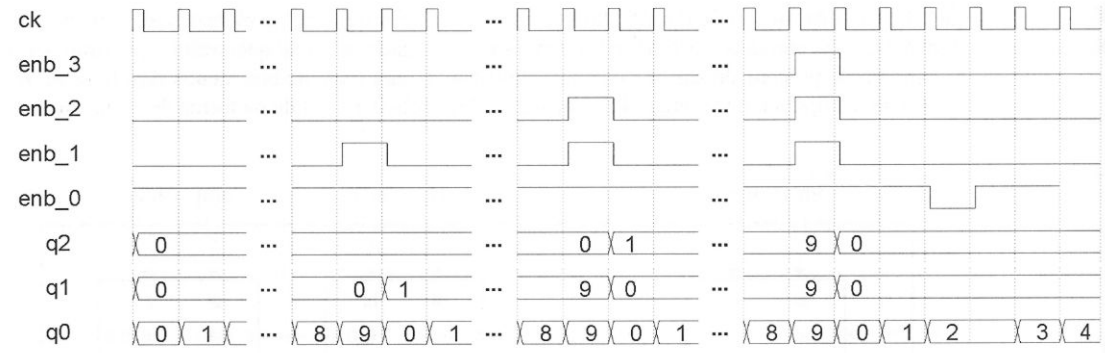


Figura 7.6.4 Carta de tempos da operação dos contadores síncronos - Exercício 7.6.5.

7.6.6 A operação de multiplicação entre dois números inteiros positivos pode ser implementada por um conjunto de somadores completos. A Figura 7.6.5 ilustra o caso de um multiplicador com 3 bits, onde " a " e " b " são os sinais de entrada, e " p " é o sinal de saída. Apresente a descrição do multiplicador proposto.

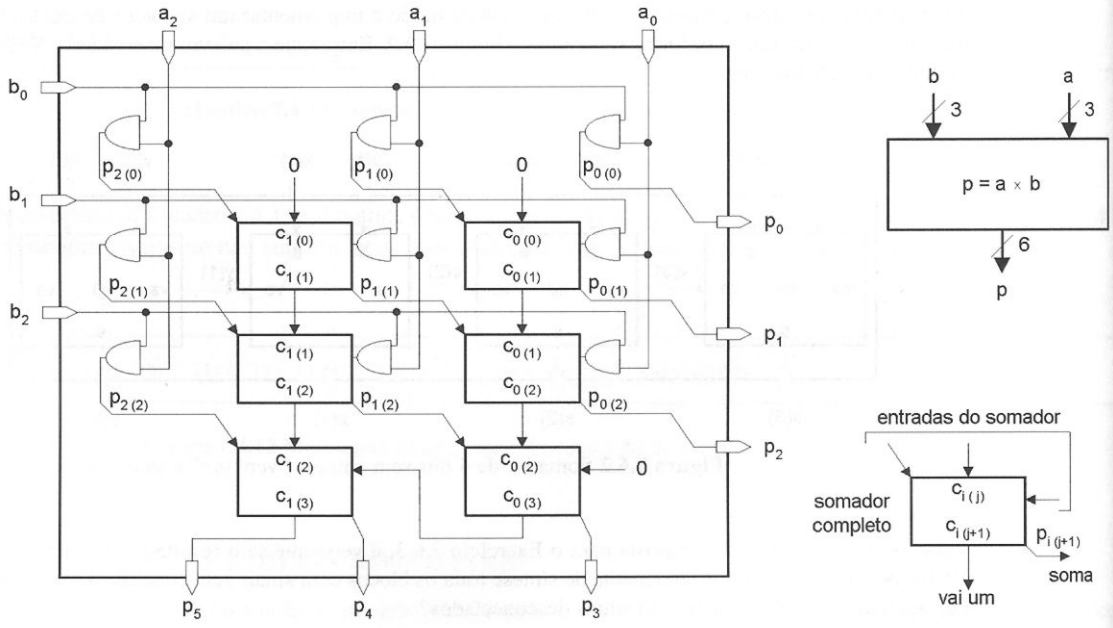


Figura 7.6.5 Multiplicador celular para operandos de três bits.

7.6.7 Altere a descrição " rev_bit0 " (Quadro 7.4.3), para empregar o comando " LOOP " com o esquema de iteração " WHILE ".

7.6.8 Apresente uma rotina de conversão de um valor do tipo inteiro para o tipo vetor de bits empregando o comando " NEXT ".

7.6.9 O circuito da Figura 7.6.6 é composto por dois blocos denominados " bloco1 " e " bloco2 ". O bloco 1 permite armazenar 4 dados em um conjunto de registradores, e o bloco 2 apresenta o valor dos registradores em 4 mostradores de 7 segmentos. A entrada dos dados é feita pela linha " de ", e a seleção do registrador que armazena os dados é feita pela entrada " end_de ". Um dado é armazenado na borda de subida do sinal " wr_l ", conforme o endereço de seleção " end_de ". Na saída " qs " do bloco 1 é possível ler o valor armazenado nos registradores conforme o sinal de seleção " end_qs ".

No bloco 2, um contador interno gera um sinal de varredura selecionando periodicamente um dos quatro mostradores através das linhas " mst ". Conjuntamente com o acionamento do mostrador, o sinal de seleção endereça o registrador correspondente ao mostrador através das linhas " end_qs ". O valor do registrador é decodificado para 7 segmentos e transferido para as saídas " sgm ". A base de tempo " ck " determina a freqüência de varredura dos mostradores.

Considere todos os sinais do tipo " bit " ou " bit_vector " e o sinal " end_qs " do tipo inteiro. Apresente a descrição do circuito completo contendo os dois blocos na forma de componentes.

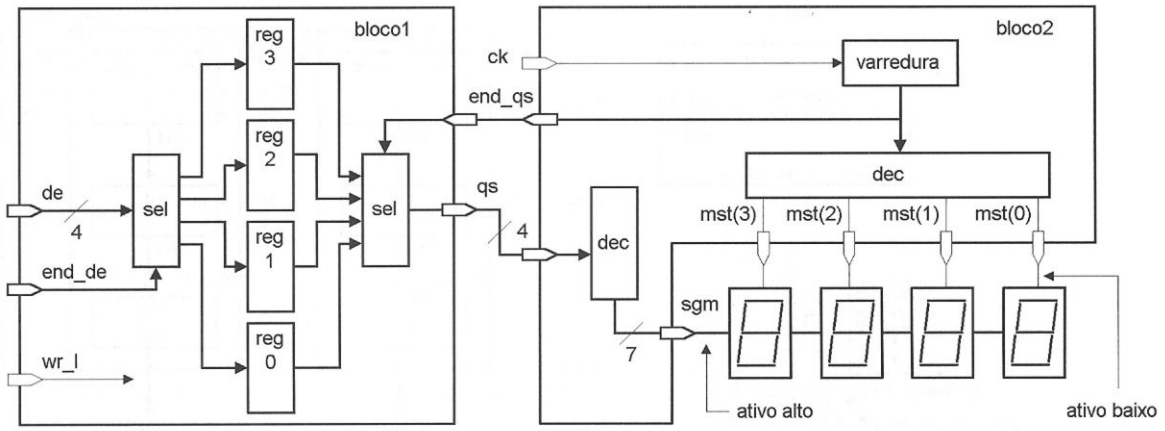


Figura 7.6.6 Diagrama de blocos do Exercício 7.6.9.

7.6.10 O objetivo deste exercício é gerar um controlador para um conversor analógico digital do tipo dupla rampa. A Figura 7.6.7 ilustra a operação do sistema. O valor da tensão a ser medido, " Vm ", é aplicado a um circuito integrador por um período de tempo fixo " tx ". Após esse período de tempo, uma tensão de referência, " Vref ", de polaridade oposta é aplicada à entrada do integrador, e um ciclo de contagem é iniciado. Quando a tensão na saída do integrador, " Vint ", atingir o valor zero, o processo de contagem é interrompido. O valor resultante dessa contagem é proporcional à tensão medida.

O diagrama do controlador a ser projetado é apresentado na Figura 7.6.8. Ele é composto por uma máquina de estados que gera os sinais de controle, contadores decimais, registradores e conversores de código " bdc " para 7 segmentos. Os contadores executam a contagem dos períodos " tx " e " tm ", e são equivalentes aos propostos no Exercício 7.6.5. Os registradores armazenam o valor da última contagem, permitindo, assim, a visualização da última medida nos mostradores de 7 segmentos. A operação deve iniciar com o acionamento da chave " ch_vm ", e a máquina de estados deve aguardar o final da contagem, sinal " enb_3=1 ". Após esse período de tempo, " tx ", a chave " ch_ref " deve ser acionada, e a máquina de estados deve aguardar a tensão na saída do integrador atingir o valor zero, sinal " Vint_z=1 ". Uma vez " Vint_z=1 ", o dado dos contadores deve ser transferido para os registradores no próximo ciclo de relógio, atualizando os mostradores com um novo valor de tensão. A operação termina com o acionamento da chave " ch_zr " para levar a saída do integrador a zero, e a inicialização dos contadores com o valor zero. Observe que, ao longo de toda a conversão, apenas uma das chaves é acionada.

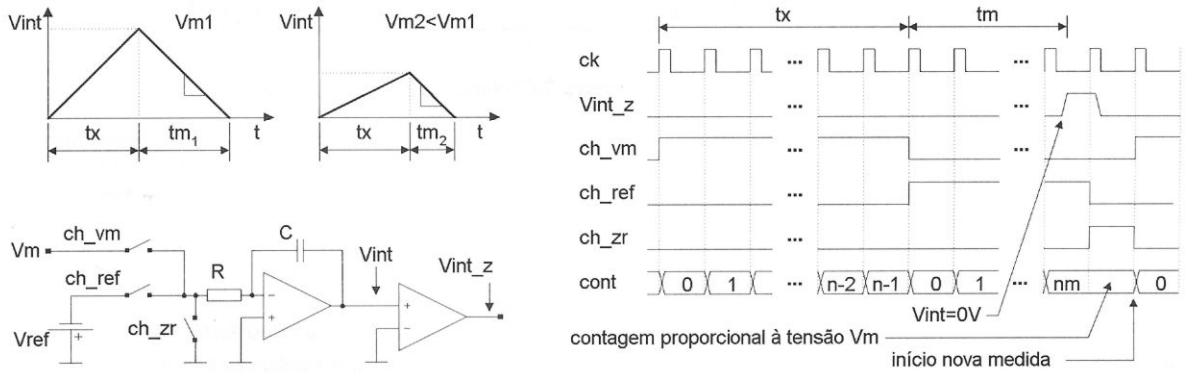


Figura 7.6.7 Conversor analógico digital dupla rampa - Exercício 7.6.10.

Considere a implementação com três níveis de hierarquia. O nível mais baixo contém a descrição do contador síncrono, registrador e decodificador " BCD " para 7 segmentos. O segundo nível, composto por um contador, um registrador e um decodificador, corresponde ao " bloco1 " da Figura 7.6.8. O nível mais elevado contém a máquina de estados finitos e três unidades " bloco1 ". No esquema proposto foi inserido o sinal " inicio " para inicializar as operações da máquina de estados. Como sugestão, operação de carga dos registradores é comandada pelo sinal " vint_z ". Isso é possível, considerando-se que, no ciclo de relógio após a condição " vint_z=1 " ser detectada, a máquina deve acionar a chave " ch_zr ", levando a tensão na saída do integrador a zero, e, portanto, o sinal " vint_z " deve, também, ir a zero. Os elementos contidos nos retângulos pontilhados são externos.

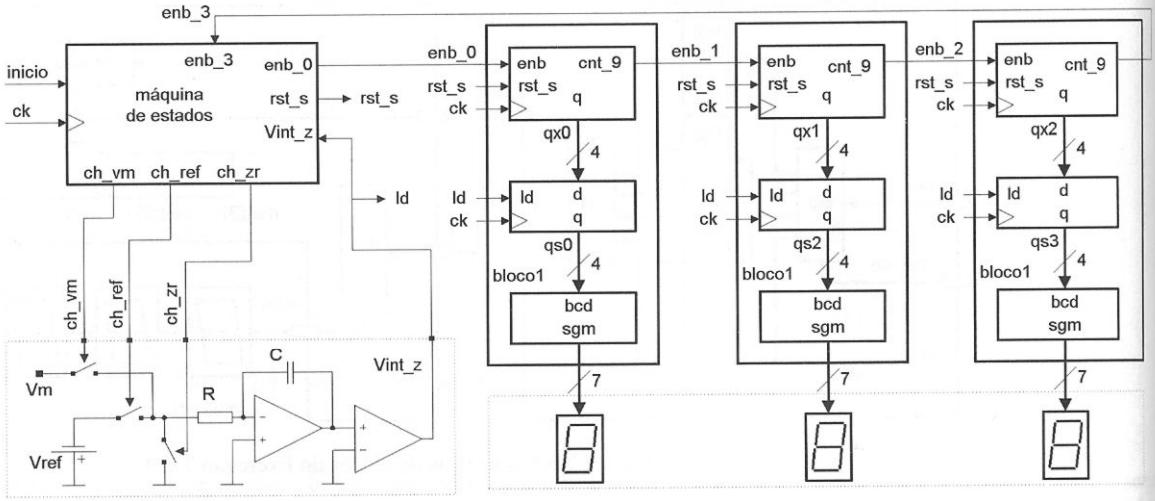
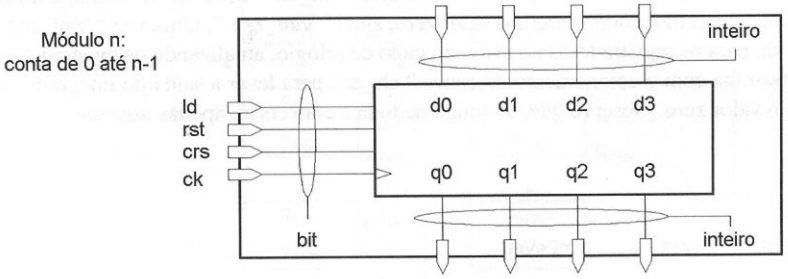


Figura 7.6.8 Diagrama de blocos do controlador para o conversor - Exercício 7.6.10.

7.6.11 Apresente a descrição de um contador com 4 bits crescente/decrescente com opção para carga de dados paralela e *reset* assíncrono. O módulo do contador deve ser definido por uma cláusula " GENERIC ", permitindo, assim, a alteração do valor máximo atingido pelo contador. O valor máximo de contagem pode estar na faixa de oito a quinze. A Figura 7.6.9 ilustra o contador. O terminal " ld " carrega o 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 ".



crs	rst	ld	q3	q2	q1	q0
X	1	X	0	0	0	0
X	0	1	d3	d2	d1	d0
1	0	0	cont. decresc.			
0	0	0	cont. cresc.			

Figura 7.6.9 Ilustração para o Exercício 7.6.11.