Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Departamento de Engenharia de Computação e Automação
DCA0212.2 - Circuitos Digitais
Docente: Tiago Barros

Laboratório 06 - Contadores

Objetivos

1. Experimentar a descrição em VHDL de circuitos na forma comportamental;
2. Reforçar os conceitos de contadores assíncronos e síncronos;
3. Pôr em prática conceitos aprendidos na disciplina Circuitos Digitais - Teoria.

Introdução

Na aula de hoje serão apresentados os blocos contadores. Os contadores são divididos em contadores síncronos e assíncronos. Os contadores síncronos são aqueles em que a mudança do seu estado ocorre em cada pulso de clock, enquanto os contadores assíncronos não possuem uma entrada de clock. Nessa aula, trataremos apenas dos contadores síncronos, sendo apresentados os contadores crescentes e decrescentes.

Contadores

Um contador de *N* bits é um componente que pode incrementar ou decrementar o próprio valor a cada ciclo de relógio, quando uma entrada de habilitação é 1.

Incrementar significa adicionar 1 a sua contagem e decrementar significa subtrair 1 a sua contagem. Um contador que pode incrementar é conhecido como ***contador crescente***, um contador que pode decrementar é conhecido como ***contador decrescente*** e um contador que pode incrementar e decrementar é conhecido como ***contador crescente/decrescente***.

Um contador crescente de quatro bits produz a seguinte sequência: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111, 0000, 0001, etc. Note que quando o contador atinge o seu valor mais alto (1111), ele retorna para 0.

De forma similar, um contador decrescente dá uma volta completa quando chega a 0 e, em seguida, indo para o valor mais elevado. Na **Fig. 1** é apresentado o diagrama de blocos de um contador crescente de 4 bits. Quando cnt=1, o contador incrementa seu próprio valor a cada ciclo de relógio. Quando cnt=0, o contador mantém o seu valor atual.

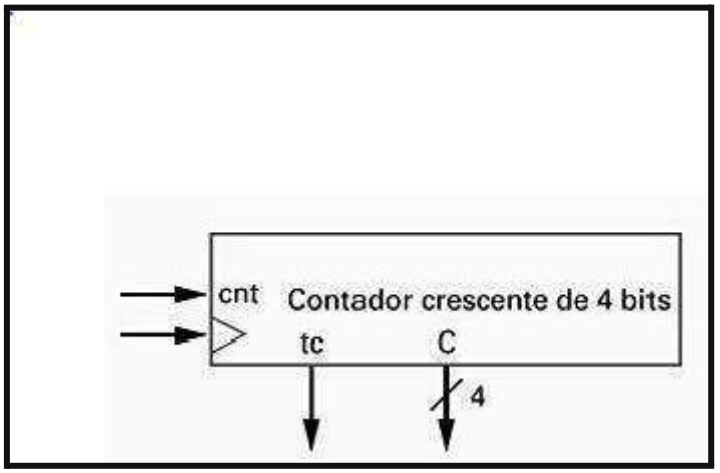

Fig. 1: Diagrama de blocos de um contador crescente de 4 bits.

Contador crescente

Um contador crescente pode ser implementado utilizado registradores. A **Fig. 2** apresenta um exemplo de implementação, o qual utiliza um bloco incrementador para somar 1 ao valor atual do registrador. Quando cnt=0, o registrador mantém o seu valor atual e, quando cnt=1, o valor corrente no registrador é somado de 1.

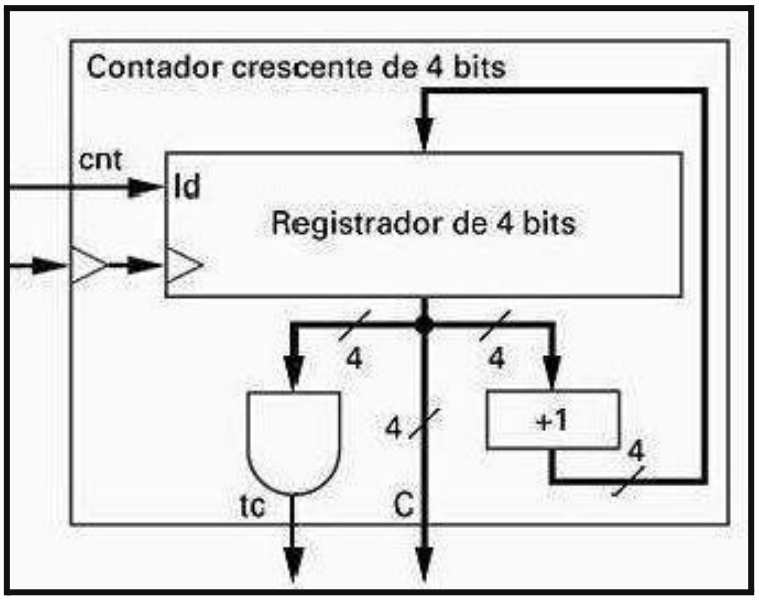

Fig. 2: Contador síncrono crescente.

A tabela-verdade de um contador crescente é apresentada na **Fig. 3**. Note que as saídas s0, s1, s2 e s3 assumem os valores dos bits de entrada. Quando o contador chega em sua contagem final, as suas saídas retornam para 0.

Entradas				Saídas				
a3	a2	a1	a0	c0	s3	s2	s1	s0
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	0	1	1
0	0	1	1	0	0	1	0	0
0	1	0	0	0	0	1	0	1
0	1	0	1	0	0	1	1	0
0	1	1	0	0	0	1	1	1
0	1	1	1	0	1	0	0	0
1	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	1	0
1	0	1	0	0	1	0	1	1
1	0	1	1	0	1	1	0	0
1	1	0	0	0	1	1	0	1
1	1	0	1	0	1	1	1	0
1	1	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	0

Fig. 3: Tabela-Verdade de um contador crescente (note que a coluna *c0* da tabela diz respeito ao *carry-out* do contador, que pode ou não estar presente no circuito).

Contador decrescente

Um contador decrescente pode ser projetado de forma semelhante a um contador crescente, substituindo o incrementador por um decrementador. Na **Fig. 4** é apresentada a estrutura de um decrementador de 4 bits.

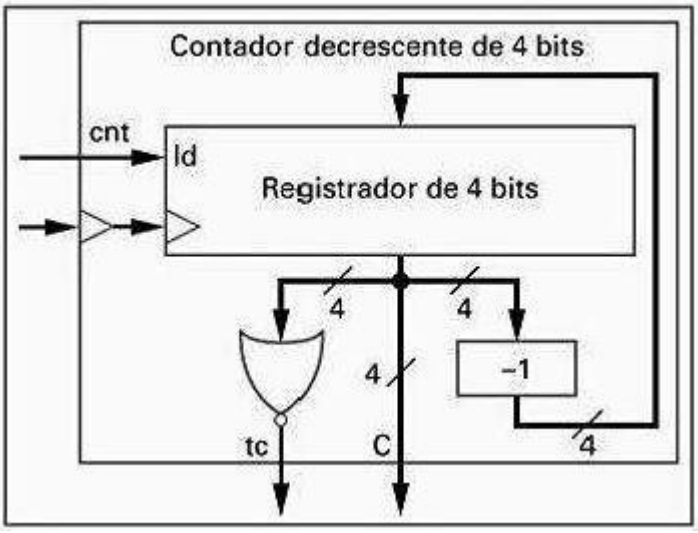

Fig. 4: Estrutura de um contador decrescente de 4 bits.

Contador com carga (*load*) paralela:

Frequentemente, os contadores vêm com a capacidade de inicialização do valor de contagem. Isto é conseguido carregando-se o registrador do contador com dados em paralelo. A **Fig. 1** mostra o projeto de um contador crescente de quatro bits com carga paralela. Quando a entrada de controle *ld* é 1, o multiplexador 2 x 1 deixa passar a entrada L com os dados a serem carregados; quando *ld* é 0, o multiplexador deixa passar o valor incrementado. Além disso, fazemos uma operação OR com os sinais *ld* e *cnt* do contador para gerar o sinal de carga do registrador. Quando *cnt* é 1, o valor incrementado é carregado. Quando *ld* é 1, os dados da carga paralela são carregados. Mesmo que *cnt* seja 0, *ld* = 1 fará o registrador ser carregado. De modo semelhante, um contador decrescente ou um crescente/decrescente poderia ser ampliado para permitir carga paralela.

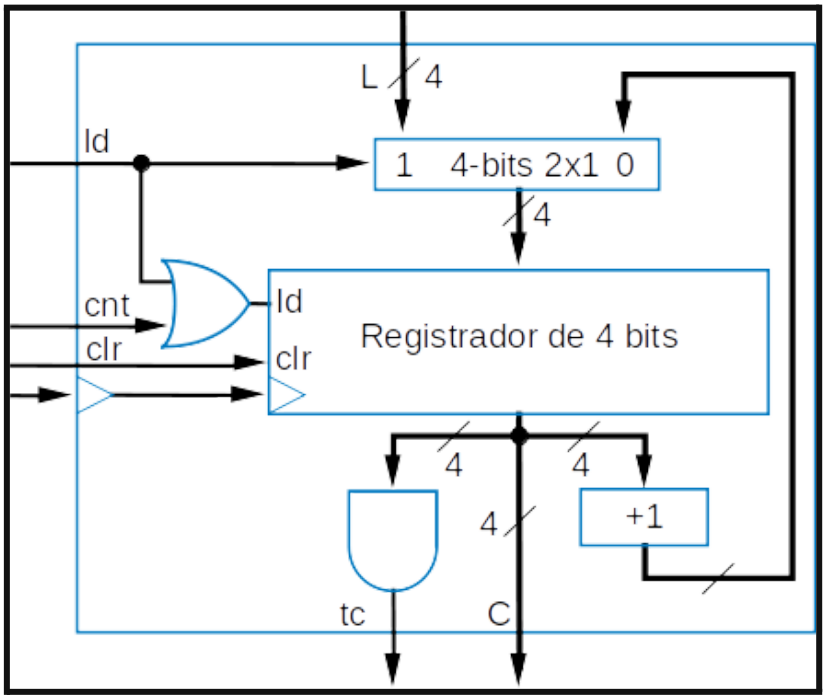

Fig. 5: Estrutura de um contador crescente de 4 bits com carga paralela.

O **Quadro 1** apresenta uma descrição comportamental de um contador com carga em paralelo, descrita em VHDL. Note que esta descrição faz uso da estrutura IF-ELSE-ELSIF da linguagem VHDL, a qual se trata de uma estrutura de execução sequencial e, portanto, deve ser descrita dentro da estrutura PROCESS da linguagem VHDL.

Quadro 1: Descrição comportamental de um contador de 4 bits com carga paralela.

```
ENTITY contador IS
PORT(
  clk:    IN BIT; --entrada de clock
  ld:     IN BIT; --carrega os dados
  reset: IN BIT;
  data:  IN INTEGER RANGE 15 DOWNTO 0; -- entrada de dados
  q:      OUT INTEGER RANGE 15 DOWNTO 0); --saída de dados
END contador;
```

```
ARCHITECTURE comportamento OF contador IS
  BEGIN PROCESS(clk,reset)
    VARIABLE qv: INTEGER RANGE 15 DOWNTO 0; --variável para a saida

    BEGIN
      IF(reset = '1') THEN
        qv := 0;
```

```
    ELSIF(clk ' event and clk = '1') THEN
      IF(ld = '1') THEN
        qv := data;
      ELSE
        IF(qv >= 9) THEN
          qv := 0;
        ELSE
          qv := qv + 1;
        END IF;
      END IF;
    END IF;
    q <= qv;
  END PROCESS;
END;
```

Atividades:

1. Implemente um contador crescente de 4 bits que conte até sua última contagem e retorne ao seu valor inicial. Adicione o sinal indicativo de término de contagem: tc=1 quando o contador atingir 1111 e tc=0 caso contrário.
2. Implemente um contador decrescente de 4 bits. Adicione o sinal indicativo de término de contagem: tc=1 quando o contador atingir 0000 e tc=0 caso contrário.
3. Satisfaça **apenas** uma das seguintes condições:
    a. Apresente o trabalho contendo os códigos e simulações em sala de aula até o dia **25/11/22** OU;
    b. Elabore e entregue um relatório com a descrição do experimento, códigos desenvolvidos e resultados de simulações que comprovem o funcionamento dos componentes desenvolvidos nos itens 1 e 2, até o dia **1/12/22**.