Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Departamento de Engenharia de Computação e Automação
DCA0212.1 - Circuitos Digitais
Docentes: Tiago Barros

# Laboratório 2 – Circuitos Combinacionais

## Objetivos:

1. Construir circuitos combinacionais complexos a partir de portas lógicas simples;
2. Testar a utilização do VHDL como ferramenta para descrever circuitos complexos fazendo uso de subprojetos.
3. Pôr em prática conceitos aprendidos na disciplina Circuitos Digitais - Teoria.

## Introdução Teórica:

Uma das descrições que o VHDL permite é a descrição estrutural. Nesta descrição todos os componentes e suas interconexões, nas quais há atribuições de sinais, são feitas através do mapeamento de entradas e saídas de componentes. Ou seja, é como se fosse uma lista de ligações entre componentes básicos pré-definidos.

Para projetos grandes ou projetos em que mais de uma pessoa está trabalhando, é viável a utilização de componentes. Um componente interliga entidades de modo a criar uma hierarquia entre elas. Outra utilidade da utilização de componentes é a não repetição de comandos que serão muito utilizados. Basicamente, um componente é dividido em declaração e instanciação.

Para utilizar um componente, é necessário que o arquivo VHDL do componente que está sendo utilizado esteja na mesma pasta onde o projeto será compilado e executado.

## Declaração:

A declaração de um componente é semelhante à declaração de uma entidade, utilizando o comando **COMPONENT** no lugar de **ENTITY**. Esta declaração deve ser feita dentro da arquitetura, logo antes do **BEGIN.**

**Exemplo:**

```
COMPONENT PortaAnd3Entradas
PORT(A,B,C : IN BIT;
        S1: OUT BIT);
END COMPONENT;
```

Em outro arquivo a porta **AND** de três entradas está definida como:

```
ENTITY PortaAnd3Entradas IS
PORT(in1,in2,in3 : IN BIT;
        Saida      : OUT BIT);
END PortaAnd3Entradas;

ARCHITECTURE behav OF PortaAnd3Entradas IS
BEGIN
  saida <= in1 and in2 and in3;
END ARCHITECTURE behav;
```

## Instanciação:

A instanciação de um componente é feita através de um rótulo, um nome e um mapeamento de portas para o componente. O mapa de portas consiste em uma lista que faz a ligação entre os sinais do componente com os sinais da entidade que solicita o componente.

Instanciação:

```
Rotulo: Nome_Componente PORT MAP(lista_pinos_componentes => lista_pinos_entity);
```

Exemplo da instanciação de um componente:

```
u1 : PortaAnd3Entradas PORT MAP (A => in1, B => in2, C => in3, S1 => saida);
```

Usualmente, os componentes são ligados por sinais na forma de fios. Esses fios são chamados de SIGNAL em VHDL. A declaração destes fios é feita da seguinte forma:

SIGNAL <nome_do_sinal>: Tipo_de_dado;

Exemplo:
SIGNAL saida1 : BIT;

A declaração do sinal deve ser feita antes do “BEGIN” da arquitetura da entidade.

# Exemplo:

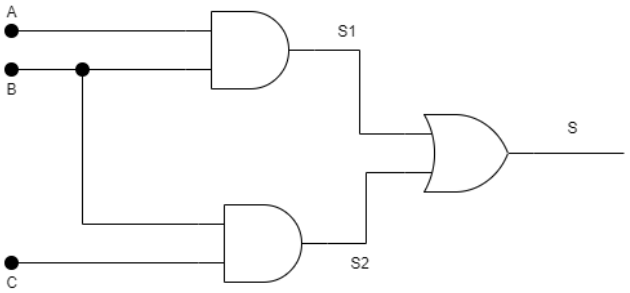


Figura 1: Circuito AB + BC

Dado o circuito apresentado na Figura 1, podemos descrevê-lo em VHDL começando pela codificação dos seguintes componentes:

```
entity PortaOr is
port(input1, input2 : in bit;
       saida_or       : out bit);
end PortaOr;

architecture behav of PortaOr is
begin
  saida_or <= input1 or input2;
end architecture behav;
```

```
entity PortaAnd is
port(en1,en2     : in bit;
       saida_and : out bit);
end PortaAnd;

architecture behav of PortaAnd is
begin
  saida_and <= en1 and en2;
end architecture behav;
```

**Quadro 1:** Declaração dos componentes que fazem parte do circuito.

A seguir, unimos os dois componentes previamente descritos em um projeto maior:

```
entity circuito is
port(A, B, C : in bit;
        S        : out bit);
end circuito;

architecture behav of circuito is
  signal S1: bit; -- Linha que recebe a saída da porta and superior
  signal S2: bit; -- Linha que recebe a saída da porta and inferior
  component PortaAnd is
    port(en1,en2         : in bit;
                saida_and : out bit);
  end component;
  component PortaOR is
    port(input1, input2       : in bit;
                        saida_or  : out bit);
```

```
  end component;
begin
        u1 : PortaAnd port map(en1 => A, en2 => B, saida_and => S1);
        u2 : PortaAnd port map(en1 => B, en2 => C, saida_and => S2);
        u3 : PortaOR  port map(input1 => S1, input2 => S2, saida_or => S);
end architecture behav;
```

**Quadro 2:** Código do circuito da Figura 1.

# Atividades

Vamos projetar um circuito que conta o número de bits iguais a 1 presente em três entradas (A, B, C) e, como saída, fornece esse número em binário, por meio de duas saídas S1 e S2.

Exemplo:

- Se a entrada for ABC = 111, então a saída deve produzir o número 3 em binário, ou seja: S1S2 = 11;
- Se a entrada for 100, então a saída deve produzir o número 1 em binário, ou seja: S1S2 = 01;
- Se a entrada for 000, então a saída deve produzir o número 0 em binário, ou seja: S1S2 = 00.

O número de 1's nas entradas pode variar de 0 a 3. Assim, uma saída com dois bits é o suficiente. Um circuito contador de 1's pode ser útil em diversas situações, como, por exemplo, estacionamentos em que sensores, localizados na parte superior das vagas e conectados a sinais luminosos, informam aos motoristas o número de vagas disponíveis em um andar específico.

Como tarefa, você deve realizar os seguintes experimentos:

1. Monte a tabela verdade do circuito, explicitando quais são as entradas e quais são as saídas e todas as possibilidades que o circuito lógico pode valer.

2. A equação do circuito é:

$$S1 = A'\cdot B\cdot C + A\cdot B'\cdot C + A\cdot B\cdot C' + A\cdot B\cdot C$$

$$S2 = A'\cdot B'\cdot C + A'\cdot B\cdot C' + A\cdot B' \cdot C' + A\cdot B\cdot C$$

É possível simplificar utilizando alguns dos postulados e identidades da lógica Booleana? Se sim, mostre qual a menor equação do circuito que você consegue obter.

3. Represente os circuitos na forma de portas lógicas (caso simplifique a equação, represente o circuito simplificado).

4. Descreva o circuito em VHDL e simule-o utilizando o Quartus/Modelsim. Para utilizar as portas lógicas, crie um projeto separado para cada porta e utilize o comando "COMPONENT" e "PORT MAP".

5. Elabore e entregue um relatório contendo a execução correta dos itens 1 a 4.