



Disciplina: Laboratório de Sistemas Digitais

Professor: Ricardo de Oliveira Duarte

Estudantes: Igor Braga de Lima, Matheus Vinícius Freitas Oliveira dos Santos e Stéphanie Pereira Barbosa

Turma: PN5

Guia de aula: 05

ATIVIDADE TEÓRICA

1) Quais são os operadores lógicos em VHDL?

And, or, nand, nor, xor, xnor.

2) Quais são os operadores de relação em VHDL?

=, \=, <, <=, >, >=.

3) Quais são os operadores *shiftem* VHDL?

logical: sll - shift left logical

srl - shift right logical

arithmetic: sla - shift left arithmetic

sra - shift right arithmetic

rotate : rol - rotate left

ror - rotate right

4) Quais são os operadores *addingem* VHDL?

+, -, &.

5) Quais são os operadores *signem* VHDL?

+, -.

6) Quais são os operadores *multiplyingem* VHDL?

*, /, mod, rem.

7) Que outro tipo de operadores existe, fora dos citados acima?

Além desses operadores, há os operadores *miscellaneous*: **, abs e not.

8) Como usar os operadores em VHDL?

Usamos os operadores em operações e associações entre sinais e variáveis. Para usá-los em VHDL basta importar os pacotes que contém os operadores e aplicar aos sinais de entrada e saída.

9) Quais as regras de precedência entre operadores em VHDL?

Os tipos de operadores possuem a seguinte ordem de precedência, da menor para a maior: *logical*, *relational*, *shift*, *adding*, *sign*, *multiplying* e *miscellaneous*. No entanto, não há ordem de precedência entre os operadores de cada tipo.

10) Em qual categoria pertence o operador not e porquê?

O operador *not* pertence à categoria *miscellaneous*, isto porque ele é um operador de inversão.

11) Quais situações o operador de concatenação & é útil?

Quando precisamos associar dois sinais ou um valor e um sinal para formar um sinal maior.

ATIVIDADE PRÁTICA

1) Utilizando como base o conhecimento adquirido, descreva um circuito em VHDL que reproduza a seguinte função $f(x) = r \cdot x \cdot \text{not}(x)$, sendo $r=2$, para qualquer valor de x e assumo no máximo 4 bits de representação.

```
1  --f(x)= r*x*not(x), sendo r=2, para qualquer valor de x e assumo no máximo 4 bits de representação.
2
3  library ieee;
4  use ieee.std_logic_1164.all;
5  use ieee.numeric_std.all;
6
7  entity exercise is
8  | port(X : in std_logic_vector (3 downto 0);
9  |   SAIDA : out std_logic_vector(7 downto 0));
10 end exercise;
11
12 architecture data_flow of exercise is
13 begin
14 | SAIDA <= std_logic_vector(unsigned(X) * not(unsigned(X)) sll 1 );
15 end data_flow;
```

2) Escreva um *testbench* para a função em VHDL que você implementou.

```
1  --testbench
2
3  library ieee;
4  use ieee.std_logic_1164.all;
5
6  entity tb_exercise is
7  end tb_exercise;
8
9  architecture tb_architecture of tb_exercise is
10 | component exercise is
11 |   port (X : in std_logic_vector (3 downto 0);
12 |     SAIDA : out std_logic_vector(7 downto 0));
13 | end component;
14
15 | signal x : std_logic_vector (3 downto 0);
16 | signal saida : std_logic_vector (7 downto 0);
17
18 | begin
19 |   tb_instancia: exercise port map (X => x, SAIDA => saida);
20 |   x <= "0000", "0100" after 10 ns, "1111" after 20 ns, "0000" after 30 ns;
21 end tb_architecture;
```

3)Compile e simule.

