



Lista de Exercícios – 04: Tipos de Dados, Classes de Objetos e Operadores

1. Com relação às classes de objetos, marque a(s) alternativa(s) correta(s).
 - ☐ Variável é um valor imposto que pode ser alterado no decorrer do código.
 - ☐ Constante pode ser empregada em regiões de código sequencial.
 - ☐ Sinais são objetos que podem ter o seu valor alterado.
 - ☐ Constantes não podem ser declaradas em declaração da entidade, arquitetura da entidade, pacote, processo e subprogramas.
 - ☐ Sinais podem ser declarados na declaração da entidade e na arquitetura da entidade.
 - ☐ Variável não pode ser declarada e empregada em regiões de código sequencial.
 - ☐ Sinais são empregados em regiões de código concorrente e sequencial.
 - ☐ Constante é um objeto com um valor estático.
2. Diante das declarações abaixo, marque V (verdadeiro) ou F (falso). No caso da alternativa ser falsa, justifique.

Justificativa abaixo:

☐ CONSTANT Pi : REAL <= 3.14;
☐ CONSTANT atraso : TIME := 50 ns;
☐ VARIABLE var <= INTEGER;
☐ VARIABLE aux := BIT : '0';
☐ SIGNAL tempo : TIME := 50 ns;
☐ SIGNAL clk : BIT <= '0';
☐ CONSTANT Pi : REAL := 3.14;
☐ CONSTANT atraso <= TIME := 50 ns;
☐ VARIABLE var = INTEGER;
☐ VARIABLE aux : BIT := '0';
☐ SIGNAL tempo := TIME := 50 ns;
☐ SIGNAL clk <= BIT := '0';
☐ CONSTANT Pi := REAL : 3.14;
☐ CONSTANT atraso : TIME = 50 ns;
☐ VARIABLE var : INTEGER;
☐ VARIABLE aux = BIT = '0';
☐ SIGNAL tempo : TIME <= 50 ns;
☐ SIGNAL clk : BIT := '0';

3. Com base no código em VHDL abaixo, verifique se há ocorrência de erros. Em caso afirmativo, proponha uma correção.

ENTITY signal_ent **IS**

PORT (a, b : **IN BIT**;
s : **OUT BIT**);

END signal_ent;

Proposta de correção:

ARCHITECTURE signal_arc **OF** signal_ent **IS**

BEGIN

SIGNAL tempo : **TIME** := 50 ns;

SIGNAL clk : **BIT** := '0';

s <= a **XOR** b;

clk <= **NOT** clk **AFTER** tempo;

END signal_arc;



4. Com base no código em VHDL abaixo, verifique se há ocorrência de erros. Em caso afirmativo, proponha uma correção.

ENTITY exemplo **IS**

PORT (d0, d1, d2 : **IN BIT**;
s1, s2 : **OUT BIT**);

END exemplo;

Proposta de correção:

ARCHITECTURE teste **OF** exemplo **IS**

BEGIN

processo1 : **PROCESS**(d0, d1, d2)

VARIABLE var1 : **STD_LOGIC**;

BEGIN

var1 := d0 **AND** d1;

s1 <= var1 **OR** d2;

END processo1;

processo2 : **PROCESS**(d0, d1, d2)

SIGNAL signal1 : **BIT**;

BEGIN

signal1 <= d0 **AND** d1;

s2 <= signal1 **OR** d2;

END processo2;

END teste;

5. Considere os valores inteiros a = 7, b = -3 e c = 3. Mostre os resultados para as sentenças com operadores aritméticos em VHDL, a seguir:

a) a/c = __;

b) a/(a+b) = __;

c) (a*c)/c = __;

d) (a/c)*c = __;

e) (a+b)**c = __;

f) ABS (a) + ABS (b) = __;

g) a REM b = __;

h) a MOD b = __;

6. Considere a <= "11001". Mostre os resultados para as sentenças com operadores de deslocamento em VHDL, a seguir:

Resultado:

a) x <= a SLL 2; -- "____"

b) y <= a SLA 2; -- "____"

c) w <= a SLL -3; -- "____"

d) z <= a SRL 2; -- "____"

e) r <= a SLA -3; -- "____"

f) s <= a SRL -3; -- "____"

g) t <= a ROL 2; -- "____"

h) u <= a ROR -3; -- "____"

i) v <= a ROL -3; -- "____"

j) k <= a ROR 2; -- "____"

k) y <= a SRA 2; -- "____"

l) w <= a SRA -3; -- "____"



7. Apresente o código de uma entidade de projeto que descreva as quatro expressões lógicas a seguir. Nessas expressões, considere o operador lógico de negação com maior precedência e o operador OR com menor precedência. A declaração da entidade deve ter quatro portas de entrada, **a**, **b**, **c** e **d**, e quatro portas de saída, **s1**, **s2**, **s3** e **s4**, todas do tipo bit.

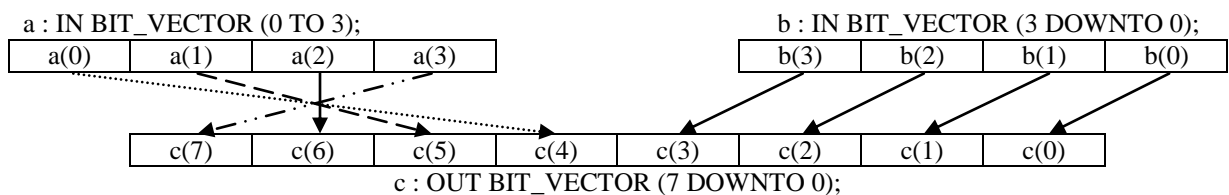
$$s1 = a + \bar{b}$$

$$s2 = a + \bar{b}.c$$

$$s3 = (a + \bar{b}).(c + d)$$

$$s4 = (a + \bar{b}).\overline{(c + a.d)}$$

8. Apresente a descrição de uma entidade de projeto (declaração da entidade mais arquitetura da entidade = código em VHDL) com duas portas de entrada e uma porta de saída, todas do tipo bit_vector. A figura a seguir ilustra o problema. Ambas as entradas possuem quatro bits, e o valor presente nessas entradas deve ser transferido para a saída, conforme ilustrado na figura.



9. Considerando a descrição ou código em VHDL a seguir, determine qual o valor de cada porta de saída. Observe que todos os comandos são concorrentes, portanto, a ordem nas linhas do código não importa.

```
ENTITY std_yc IS
  PORT (a, b : OUT BIT_VECTOR (2 DOWNT0 0);
        c, d : OUT BIT_VECTOR (0 TO 2));
END std_yc;

ARCHITECTURE teste OF std_yc IS
  CONSTANT x : BIT_VECTOR(0 TO 7) := B"1101_1001";
  SIGNAL y : BIT_VECTOR(3 DOWNT0 0);
BEGIN
  a <= x(1 TO 3);
  b <= y(3 DOWNT0 1);
  c <= x(5 TO 7);
  d <= y(2 DOWNT0 0);
  y <= x(2 TO 5);
END teste;
```



10. Considerando a descrição ou código em VHDL a seguir, identifique as linhas que contêm erros no código e proponha uma solução.

```
ENTITY errad_1 IS
  PORT (a, b, c, d : IN BIT;
        S      : OUT BIT_VECTOR (5 DOWNT0 0));
END errad_1;
```

Proposta de correção:

```
ARCHITECTURE teste OF errad_1 IS
BEGIN
  s(0) <= a AND b OR c AND d;
  s(1) <= a NOR b NOR c;
  s(2) <= a AND b OR c;
  s(3) <= NOT (a AND b) NAND c;
  s(4) <= a XOR b XOR c;
END teste;
```

11. Na descrição ou código em VHDL a seguir, para teste de operações lógicas, a declaração da entidade contém duas entradas e três saídas do tipo bit_vector. Considerando que a_bit tem o valor “010” e b_bit tem o valor “111”, determine qual o valor de cada porta de saída.

```
ENTITY std_c IS
  PORT( a_bit      : IN BIT_VECTOR(2 DOWNT0 0);
        b_bit      : IN BIT_VECTOR(2 DOWNT0 0);
        not_bit, and_bit, or_bit : OUT BIT_VECTOR(2 DOWNT0 0));
END std_c;
```

```
ARCHITECTURE exemplo OF std_c IS
BEGIN
  not_bit <= NOT a_bit;
  and_bit <= a_bit AND b_bit;
  or_bit  <= a_bit OR b_bit;
END exemplo;
```