



Lista de Exercícios - 02

1. Marque verdadeiro (V) para as palavras reservadas da linguagem VHDL e, em caso contrário, falso (F):

(V) WHEN	(V) AFTER
(V) ENTITY	(F) SET
(F) PARITY	(V) SELECT
(V) PORT	(F) REPEAT
(V) BIT	(V) TO
(V) BIT_VECTOR	(F) INTERRUPT
(F) DOWN	(F) EXTERNAL
(F) SLOW	(V) ALL
(V) IN	(V) COMPONENT
(V) BEGIN	(F) PACKED
(F) USES	(V) USE
(V) END	(F) FORWARD
(F) GOTO	(V) PACKAGE

2. Conceitue as abordagens de projeto *top-down* e *bottom-up*.

Abordagem *top-down* → abordagem em que o projeto inicia em níveis mais altos e, posteriormente, é subdividido em blocos menores.

Abordagem *bottom-up* → abordagem em que o projeto inicia-se por blocos mais simples e, posteriormente, é construído por blocos mais complexos.

3. Conceitue as palavras reservadas ENTITY e ARCHITECTURE em uma estrutura de código em VHDL. Dê um exemplo.

Declaração da Entidade (ENTITY) → definição da interface entre o componente e o ambiente exterior, isto é, definição das entradas e saídas do componente, especificadas com a declaração das portas (PORT).

Arquitetura da Entidade (ARCHITECTURE) → descrição da funcionalidade do componente, isto é, especificação do comportamento do componente em relação às suas entradas e saídas.

Exemplo:

```
ENTITY porta_and IS
    PORT (a, b : IN BIT;
          f : OUT BIT);
END porta_and;
ARCHITECTURE logica OF porta_and IS
BEGIN
    f <= a AND b;
END logica;
```



4. Dentre as características de VHDL, marque a(s) opção(ões) que estiver(em) correta(s).
- a) (F) A VHDL não é interpretada, mas sim compilada.
 - b) (V) A reusabilidade em VHDL corresponde à criação de componentes para um determinado projeto, possibilitando que esta criação possa ser reutilizada em outros projetos.
 - c) (F) Qualquer código em VHDL simulado não é associado a uma biblioteca especial ou de trabalho, na qual estão todos os componentes utilizados no projeto.
 - d) (V) A VHDL possibilita a criação de novos pacotes e bibliotecas, além das bibliotecas já existentes pré-compiladas.
5. Enumere a segunda coluna de acordo com a primeira coluna em se tratando dos modos de operação de uma porta em VHDL.
- | | |
|--------------|---|
| (1) IN | (3) porta que opera exclusivamente como saída. |
| (2) BUFFER | (4) porta pode ser de entrada e saída ao mesmo tempo. |
| (3) OUT | (1) porta que opera exclusivamente como entrada. |
| (4) INOUT | (2) porta similar à saída que pode ter realimentação interna. |
6. Identifique na Figura 1, a seguir, qual parte corresponde à declaração da entidade e qual parte corresponde à arquitetura da entidade. Descreva a sintaxe em VHDL, tanto para a declaração da entidade quanto para a arquitetura da entidade.

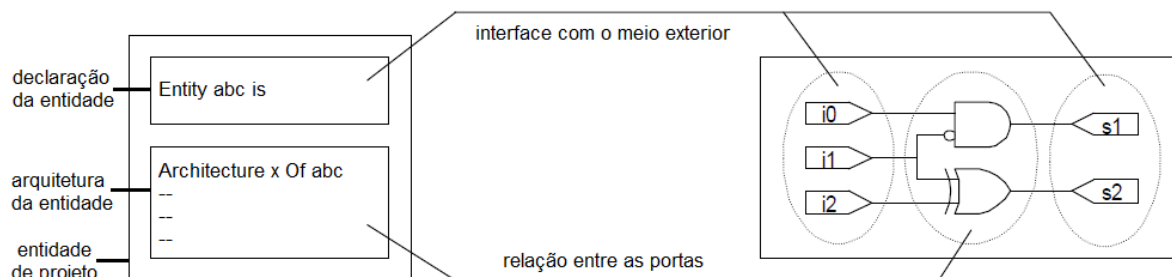


Figura 1

```
ENTITY nome IS
    GENERIC (parametro1 : tipo_dado := valor);
    PORT (entrada1, entrada2 : IN tipo_dado;
          saida : OUT tipo_dado);
END nome;
ARCHITECTURE nome_identificador OF entidade_abc IS
    --
    -- regiao de declaracoes:
    --   declaracoes de sinais e constantes
    --   declaracoes de componentes referenciados
    --   declaracao e corpo de subprogramas
    --   definicao de novos tipos de dados locais
    --
BEGIN
    --
    -- comandos concorrentes
    --
END;
```



7. Com relação ao uso de nomes das entidades, subrotinas, variáveis, constantes e sinais, marque verdadeiro (V) ou falso (F) para as afirmações abaixo.
- (V) Deve-se usar apenas caracteres alfanuméricos e o caractere underline (_).
- (V) O primeiro caractere deve ser uma letra.
- (F) Há caso sensitivo, isto é, há distinção entre letras maiúsculas e minúsculas.
- (V) NOME, Nome e nome são um mesmo identificador.
8. Marque a(s) alternativa(s) correta(s):
- a) (X) A palavra reservada PORT corresponde a uma lista com especificações de todas as portas de entradas e de saídas de um circuito.
 - b) (X) O modo de uma porta indica se ela é uma porta de entrada, de saída, bidirecional ou um buffer.
 - c) () A palavra reservada ARCHITECTURE define a interface do circuito.
 - d) () A palavra reservada ENTITY define a funcionalidade do circuito.
 - e) (X) A palavra reservada GENERIC permite a passagem de informações estáticas para uma unidade de projeto.
9. Identifique os 3 tipos de modelagem de arquiteturas, indicando (1) para Comportamental, (2) para Estrutural e (3) para Fluxo de Dados (RTL):
- a) (3) Especifica a própria expressão booleana do circuito.
 - b) (1) Descreve a funcionalidade do componente utilizando expressões e linguagem de alto nível.
 - c) (2) Descreve as interconexões entre os componentes.
10. A partir dos 3 códigos em VHDL abaixo, representando a implementação de um comparador de 4 bits, determine qual foi o tipo de modelagem de arquitetura utilizado em cada um deles. Justifique cada a sua resposta.

Código em VHDL – 01 → Tipo de modelagem de arquitetura: Fluxo de Dados

Justificativa:

Na descrição por fluxo de dados as operações lógicas são executadas concomitantemente, ou seja, ao mesmo tempo. A expressão booleana do circuito projetado é colocada depois do comando BEGIN, dentro da arquitetura da entidade.

```
LIBRARY IEEE;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_unsigned.all;

ENTITY comp4 IS
    PORT ( a : IN std_logic_vector (3 DOWNTO 0);
          b : IN std_logic_vector (3 DOWNTO 0);
          igual : OUT std_logic);
END ENTITY comp4;

ARCHITECTURE logica_1 OF comp4 IS
BEGIN
    igual <= '1' WHEN (a=b) ELSE '0';
END ARCHITECTURE logica_1;
```



Código em VHDL – 02 → Tipo de modelagem de arquitetura: Comportamental

Justificativa:

Basicamente, a diferença entre a arquitetura com descrição por fluxo de dados e descrição comportamental é a declaração **PROCESS** que define os processos concorrentes. No código em VHDL abaixo, é utilizado um nome opcional para o processo (comp), seguido por dois pontos, o comando ou palavra reservada **PROCESS** e entre parênteses a lista das variáveis de entrada, cuja alteração pode levar a mudanças na variável de saída (lista de sensibilidade). O processo reagirá às mudanças ocorridas na lista de sensibilidade. É importante que a lista de sensibilidade contenha todas as variáveis que possam modificar o comportamento do processo. Ressalta-se ainda que os comandos dentro de um processo são executados de forma sequencial.

```
LIBRARY IEEE;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_unsigned.all;

ENTITY comp4 IS
    PORT ( a : IN std_logic_vector (3 DOWNTO 0);
          b : IN std_logic_vector (3 DOWNTO 0);
          igual : OUT std_logic);
END ENTITY comp4;

ARCHITECTURE logica_2 OF comp4 IS
BEGIN

    comp : PROCESS (a, b)
    BEGIN
        IF a=b THEN
            igual <= '1';
        ELSE
            igual <= '0';
        END IF;

    END PROCESS;

END ARCHITECTURE logica_2;
```



Código em VHDL – 03 → Tipo de modelagem de arquitetura: Estrutural

Justificativa:

A descrição estrutural da arquitetura do comparador de 4 bits define o sinal **x** de interconexão entre os componentes XNOR ($x(0)$, $x(1)$, $x(2)$ e $x(3)$). Esse sinal é interno por isso não é listado na declaração da entidade. É definida a instanciação dos componentes XNOR e AND, na qual é possível verificar as interconexões entre as entradas, sinal e saída. Por motivos didáticos, neste exercício, os componentes não foram predefinidos ou criados e nem declarados no código em VHDL abaixo.

```
LIBRARY IEEE;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_unsigned.all;

ENTITY comp4 IS
    PORT ( a : IN std_logic_vector (3 DOWNT0 0);
          b : IN std_logic_vector (3 DOWNT0 0);
          igual : OUT std_logic);
END ENTITY comp4;

ARCHITECTURE logica_3 OF comp4 IS
    SIGNAL x : std_logic_vector (3 DOWNT0 0);
BEGIN
    u0 : XNOR PORT MAP (a(0), b(0), x(0));
    u1 : XNOR PORT MAP (a(1), b(1), x(1));
    u2 : XNOR PORT MAP (a(2), b(2), x(2));
    u3 : XNOR PORT MAP (a(3), b(3), x(3));
    u4 : AND4 PORT MAP (x(0), x(1), x(2), x(3), igual);

END ARCHITECTURE logica_3;
```