



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: 09

ATIVIDADE TEÓRICA

1) Por que o modelo de descrição estrutural em VHDL, na sua essência, pode ser considerado um modelo de descrição híbrido?

Porque o modelo de descrição estrutural consegue unir os modelos *data-flow* e *behavioral*.

2) Se temos que descrever um sistema digital modular e/ou hierárquico, a qual modelo de descrição devo adotar?

O modelo de descrição estrutural

3) Quais as vantagens de descrições modulares em VHDL?

As descrições modulares proporcionam uma maior clareza e melhor compreensão do circuito descrito uma vez que longos trechos de código podem ser substituídos módulos com nomes menores e mais representativos. Além disso, o uso de módulos economiza tempo ao evitar a reescrita de código.

4) Quais recursos da linguagem VHDL são usados para escrever descrições modulares?

Packages, components e functions.

5) Em VHDL componentes estão necessariamente associados à o que?

A uma *entity* e sua *architecture* correspondente.

6) Qual é a sequência de passos para usar um componente em VHDL?

1. Nomear o módulo a ser descrito (criação da *entity*);
2. Descrever o que o módulo irá fazer (criação da *architecture*);
3. Declarar o componente;
4. Instanciar/realizar o *mapping* do módulo no código.

7) Qual são os passos do procedimento de descrição de um sistema digital no modelo estrutural em VHDL?

1. Fazer a declaração da *top-level entity*;

2. Declarar as *design units* de nível inferior, dentro da *architecture*, que serão usadas no circuito;
3. Declarar os sinais internos usados para conectar as *design units*;
4. Instanciar as *design units*.

8) Qual a dica que o livro te dá para declarar um componente dentro de uma arquitetura *top-level* em VHDL?

Fazer uma cópia da *entity* a ser utilizada, substituir a palavra-chave *entity* por *component* e utilizar *end component* ao final da declaração.

9) O que é uma instância de um componente?

É um objeto que herda seu tipo e características de um *component*.

10) Quais são os passos do procedimento para instanciar um componente dentro de uma arquitetura descrita estruturalmente?

1. Declarar os componentes;
2. Instanciar os componentes;
3. Atribuir um *label* a instanciiação;
4. Fazer o mapeamento dos sinais do componente com ao da entidade de alto nível.

11) Que palavras reservadas da linguagem VHDL são usadas no procedimento de instanciação de um componente?

Port map.

12) Qual a boa prática de descrição em VHDL sugerida pelo livro quando se instancia um componente? E por que é considerada uma boa prática?

O uso de *labels* para todas as *design units*. Isso é considerado uma boa prática porque melhora a compreensão do código.

13) Qual a diferença entre *direct mapping* e *implied mapping*?

No *direct mapping* cada *signal* é listado e associado com o sinal que ele se conecta na *design unity* de alto nível através do uso do operador “=>”. Por sua vez, no *implied mapping* a associação é feita pela ordem que os *signals* aparecem na declaração de mapeamento da *design unity* de alto nível.

14) O que deve ficar à esquerda do operador de asserção => em um *port map*?

Os *signals* do componente instanciado.

15) O que deve ficar à direita do operador de asserção => em um *port map*?

Os *signals* correspondentes da *entity* de alto nível ou *signals* declarados dentro da *architecture*.

16) Para que serve a declaração *generic* em VHDL?

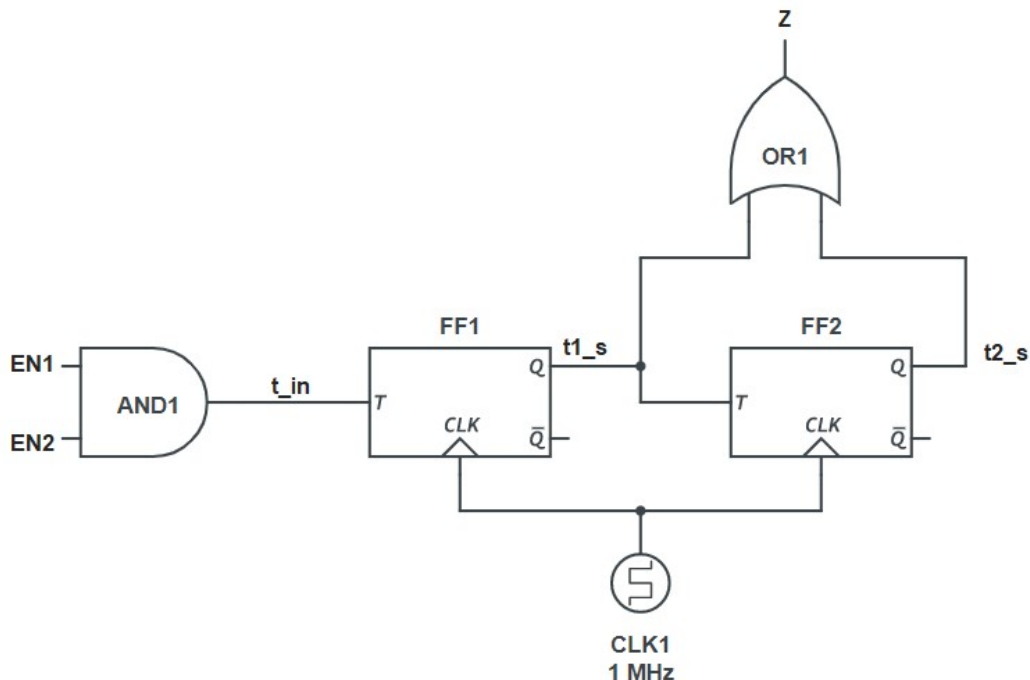
Para criar variáveis genéricas que terão seus valores definidos na instanciação do componente, sendo uma declaração prática para a criação de componentes reutilizáveis/parametrizáveis.

17) Para que serve a declaração *generic map* em VHDL?

Para definir os valores das variáveis genéricas no momento da instanciação do componente.

ATIVIDADE PRÁTICA

1) Faça um dos itens do Exercício 1 da página 131 do livro. Faça o desenho em salve em um arquivo pdf.



2) Faça um dos itens do Exercício 3 da página 132 do livro.

3) Escreva um *testbench* para o sistema em VHDL que você implementou no item anterior.

4) Compile, simule e verifique o comportamento do seu circuito.