

PSI-3451 Projeto de CI Lógicos Integrados

Aula 4

A parte prática da aula 4 pretende colocar o aluno em contato com mais algumas estruturas sintáticas da linguagem VHDL, particularmente funções e procedimentos, além de aspectos de bibliotecas externas (IEEE) e construídas pelo usuário.

Tenha certeza que sabe responder as perguntas listadas abaixo após a observação dos arquivos ou dos resultados de simulações.

PARTE PRÁTICA

São fornecidos 2 arquivos VHDL do módulo *num_gen* que faz parte do projeto **snake**. Cada um contém modelos diferentes dos *muxes* encadeados (ver apostila de conceitos). Cada um dos módulos será modificado durante a aula gerando outros arquivos. No total serão analisados e simulados 5 casos. A seguir vamos descrever cada um dos modelos e os conceitos de VHDL que podem ser observados em cada uma das simulações.

Em cada caso o procedimento para **capturar, compilar e simular** cada arquivo VHDL é o mesmo utilizado nas aulas anteriores.

Preparação das Pastas e Arquivos para Simulação

- Ligar o computador e entrar no ambiente **Windows**.
- Utilizando o **Windows Explorer** acesse a sua área de trabalho na unidade de rede **X**.
- Na sua área de trabalho crie uma pasta **X:\psi3451\aula_4** para armazenar os resultados desta prática. Como serão realizadas 5 simulações, recomendamos que sejam criadas pastas separadas para cada uma delas com o objetivo de salvar adequadamente os resultados. Por exemplo, crie as pastas:

1. X:\psi3451\aula_4\ng_1

(arquivo *num_gen_1.vhd* ; entidade *num_gen_direct*)

2. Para estudar tipos IEEE e conversões

X:\psi3451\aula_4\ng_1m

(arquivo *num_gen_1m.vhd* - inicialmente cópia do *num_gen_1* ; entidade *num_gen_direct*).

3. Para estudar o uso de funções

X:\psi3451\aula_4\ng_2

(arquivo *num_gen_2.vhd* ; entidade *num_gen_with_function*).

4. Para estudar o uso de procedimentos

X:\psi3451\aula_4\ng_3

(arquivo *num_gen_3.vhd* - inicialmente cópia do *num_gen_2* ; entidade

num_gen_with_procedure).

5. Para estudar o uso de pacotes (packages) e bibliotecas (library)

X:\psi3451\aula_4\lib_1

(arquivo *procedure_pack.vhd* - inicialmente cópia do *num_gen_3.vhd* ; pacote *procedure_pack*)

e

X:\psi3451\aula_4\ng_4

(arquivo *num_gen_4.vhd* - inicialmente cópia do *num_gen_2* ; entidade *num_gen_with_procedure_2*).

- No **Windows Explorer**, selecione **Rede=> NEWSERVERLAB => psi3451 => aula_4**.
- Copie os arquivos *num_gen_1* e *num_gen_2* para as pastas correspondentes, **X:\psi3451\aula_4\ng_1** e **X:\psi3451\aula_4\ng_2**.

1) Captura, compilação e simulação do circuito *num_gen_direct* (muxes + número direto)

- a) Abra o arquivo *num_gen_1.vhd* para a pasta **X:\psi3451\aula_4\ng_1**. Ele está codificado no modelo VHDL comportamental (*dataflow*).
- b) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido cada uma das construções abaixo (conceitos VHDL):
 - o parâmetro *generic width*
 - a inferência dos três muxes (ver [apostila de conceitos](#))
 - pacote *std_logic_1164* (ieee) e tipos *std_logic* e *std_logic_vector*
 - pacote *numeric_std* (ieee) e tipo *signed/unsigned*
 - função de conversão *to_unsigned (int, natural)*
 - type casting *std_logic_vector (unsigned)*
 - o local da atribuição de valor por **rand_num_direct** (veja onde você deverá completar os dados)
- c) Complete ao arquivo *num_gen_1.vhd* com os dados solicitados para o número aleatório (baseados no seu número USP).
- d) Faça o procedimento de captura, compilação e simulação.

Recomendação: realize a simulação para verificar:

- o funcionamento do circuito de acordo com a tabela da apostila de conceitos
- guarde os resultados do Wave para comparação na seção seguinte.

Perguntas: seguindo as recomendações, a simulação mostrou o comportamento esperado do circuito como descrito na [apostila de conceitos](#)?

2) Captura, compilação e simulação do circuito num_gen_direct no modelo dataflow, com alterações de sintaxe

- Faça uma cópia do arquivo *num_gen_1.vhd* e salve-o com o nome *num_gen_1m.vhd* na pasta **X:\psi3451\aula_3\ng_1m**.
- Em seguida modifique o arquivo (com o programa **Notepad++**), COMENTANDO a linha onde se declara o uso do pacote `std_logic_1164`.
- Faça o procedimento de captura, compilação e simulação.

Perguntas: O que ocorreu na compilação? Houve erro? Se positivo, qual foi o erro apontado pelo compilador? Por que?

- Partindo novamente do arquivo *num_gen_1.vhd* (ATENÇÃO: antes de ser modificado) repita os itens a) b) e c) fazendo as modificações listadas a seguir (uma de cada vez) e responda sempre a mesma pergunta acima.

☆ COMENTAR a linha onde se declara o pacote `numeric_std`

☆ SUBSTITUIR a linha

```
{rand_num_direct    <=    (to_unsigned    (YOUR_NUMBER_DECIMAL,
number'length));}
```

por

```
{ rand_num_direct <= YOUR_NUMBER_DECIMAL;}
```

☆ SUBSTITUIR a linha

```
{rand_num_direct    <=    (to_unsigned    (YOUR_NUMBER_DECIMAL,
number'length));}
```

por

```
{ rand_num_direct <= "YOUR_NUMBER_BINARY "};
```

☆ na linha de comando de atribuição de `one_gen_s` SUBSTITUIR `{std_logic_vector (rand_num_direct)}`

por

```
{ rand_num_direct}
```

Recomendação: Realize uma simulação com a mesma sequência de estímulos adotada na Seção 1.

3) Captura, compilação e simulação do circuito num_gen_2 (muxes + função)

Funções devem ser vistas como estruturas que objetivam estruturar a descrição do modelo e aumentar a legibilidade do VHDL. Há uma forte semelhança entre funções da linguagem VHDL com as existentes em linguagens de programação de software como C ou Pascal. Não deve ser confundido com componentes que são representações estruturais de módulos.

Como características principais, as funções:

- retornam um valor;
- são chamadas dentro de expressões;
- não modificam valores que lhe são passados;
- são executados em um ciclo de simulação, ou seja, todos os parâmetros passados são

vistos como variáveis.

- a) Abra o arquivo *num_gen_2.vhd* para a pasta **X:\psi3451\aula_4\ng_2**.
- b) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
 - o parâmetros *generic width*
 - a inferência dos três muxes como ilustrado na apostila de conceitos
 - a declaração de função **rand_num_f** (veja onde você deverá completar os dados)
 - o corpo da função
 - a chamada de função
 - a declaração de atraso inercial
- c) Complete ao arquivo *num_gen_2.vhd* com os dados solicitados para a função (baseados no seu número USP)
- d) Faça o procedimento de captura, compilação e simulação.

Recomendação: realize a simulação para verificar:

- o funcionamento do circuito de acordo com a tabela da apostila de conceitos

Guarde os resultados do Wave para comparação na seção seguinte.

Perguntas: seguindo as recomendações, a simulação mostrou o comportamento esperado do circuito como descrito na apostila de conceitos? O tempo de atraso especificado foi observado na simulação?

4) Captura, compilação e simulação do circuito num_gen_2 (muxes + procedimento)

Procedimentos também são estruturas que objetivam estruturar a descrição do projeto e aumentar a legibilidade do VHDL. Um procedimento pode ser visto como a descrição de um processo que pode ser invocado várias vezes. Um procedimento, diferente de uma função, tem como características principais:

- pode gerar múltiplos valores, porém, não pelo comando de retorno;
- pode ter comando de retorno como forma de encerrar um procedimento;
- é chamado como comando;
- pode modificar valores que lhe são passados como parâmetros;
- apresentam noção de tempo.

- a) Faça uma cópia do arquivo *num_gen_2.vhd* e salve-o com o nome *num_gen_3.vhd* na pasta **X:\psi3451\aula_4\ng_3**.
- b) Como forma de experimentar diversas estruturas em VHDL, você deve trocar a função por um procedimento. Usando os slides da aula para suporte para a sintaxe de procedimento:
 - denomine a entity de **num_gen_with_procedure**
 - denomine o procedimento de **rand_num_p**
 - verifique como compatibilizar a sintaxe de chamada de procedimento para a sintaxe do terceiro mux.
 - deixe o tempo de atraso especificado dentro do procedimento

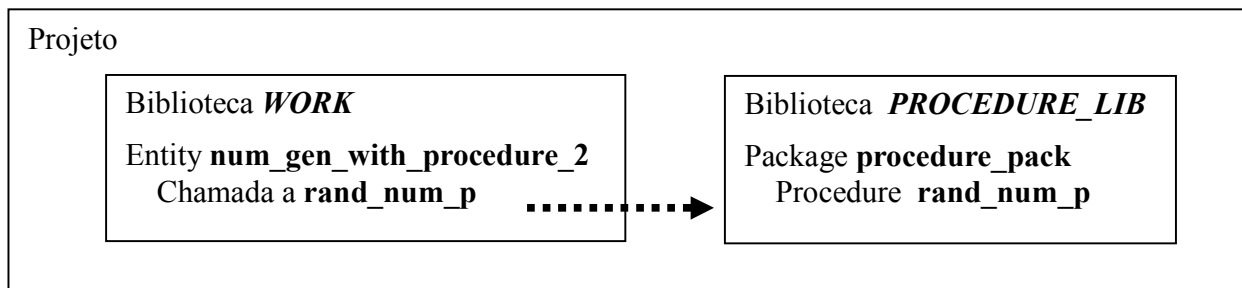
c) Faça o procedimento de captura, compilação e simulação.

Guarde os resultados do Wave para comparação na seção seguinte.

Perguntas: a simulação mostrou o comportamento idêntico ao do item 3)? O tempo de atraso especificado foi observado na simulação?

5) Captura, compilação e simulação do circuito `num_gen_with_procedure` com procedimento declarado em pacote de biblioteca específica

Nesta seção, o procedimento `rand_num_p` será colocado em uma biblioteca que pode ser acessada publicamente por descrições de circuito em um sistema de arquivo separado. Para isto, ele será declarado dentro de pacote `procedure_pack` a ser instalado em uma biblioteca `procedure_lib`. Neste experimento, o procedimento será chamado por um circuito, cuja unidade de projeto estará em *work*, como mostrado no diagrama abaixo.



Primeiramente, **a nova biblioteca e o pacote** devem ser construídos. O package deve ser compilado em uma biblioteca *work* (mapeada para pasta específica) e depois remapeada com o nome `procedure_lib`.

- O pacote será editado a partir do arquivo `num_gen_3.vhd`. Faça uma cópia deste e salve-o com o nome `procedure_pack.vhd` na pasta `X:\psi3451\aula_4\lib_1`.
- Edite o arquivo `procedure_pack.vhd`:
 - Crie o pacote `procedure_pack` com a sintaxe de package (declaração e *body*). Use os slides da aula 3 para se referenciar.
 - O único elemento do pacote será o procedimento `rand_num_p`
- No Modelsim, acesse o diretório `X:\psi3451\aula_4`.
- Faça o procedimento de criação de nova biblioteca na opção **Create**, selecione "**A new library and a physical mapping to it**". Use `lib_1` em ambos os campos.
- Mude o diretório para `X:\psi3451\aula_4\ng_4`.
- Crie uma nova biblioteca e na opção **Create**, selecione "**a map to an existing library**". Deixe *work* como nome e mapeie-o para `X:\psi3451\aula_4\lib_1` para o seu nome físico.
- Faça ali o procedimento de compilação do arquivo `procedure_pack.vhd`.
- Clique o botão direito sobre *work* e edite. Mude o nome da biblioteca para `procedure_lib`.

Pergunta: Qual é a biblioteca estabelecida e o que ela contém?

Nos passos seguintes, a nova biblioteca e o recém-criado pacote serão utilizados pelo novo módulo *num_gen_4*.

- i) Faça uma cópia do arquivo *num_gen_3.vhd* e salve-o com o nome *num_gen_4.vhd* na pasta **X:\psi3451\aula_4\ng_4**.
- j) No Modelsim, edite o arquivo *num_gen_4.vhd* da seguinte forma:
 - denomine a entity como **num_gen_with_procedure_2**
 - elimine todas as declarações de **rand_num_p** (obs. a chamada ao procedimento dentro da arquitetura deve ser mantida)
 - acrescente a declaração da biblioteca **procedure_lib** e ao uso do pacote **procedure_pack**
- k) Faça o procedimento de criação de biblioteca *work* e na opção **Create**, selecione "**A new library and a physical mapping to it**".
- l) Faça o procedimento de compilação de e simulação do arquivo *num_gen_4.vhd*.

Pergunta: a simulação mostrou o comportamento idêntico ao do item 4)?

Pergunta: Entendeu o uso da biblioteca criada e o seu pacote pelo circuito de muxes encadeados?

Pergunta: De que maneira a atual forma de usar pacotes é diferente daquelas usadas na aula 3 (com *reg_bank_simplicado*)? Tenha certeza que sabe diferenciar e explicar.